UNC-5157

UNC-SAM-2: A FORTRAN MONTE CARLO PROGRAM TREATING TIME-DEPENDENT NEUTRON AND PHOTON TRANSPORT THROUGH MATTER

E. S. Troubetzkoy

September 1966

Work Performed on UNC Project 2336
under New York Procurement District
Contract No. DA-30-069-AMC-785(X) for
The United States Army,
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

UNITED NUCLEAR CORPORATION
RESEARCH AND ENGINEERING CENTER

UNC-5157

UNC-SAM-2: A FORTRAN MONTE CARLO PROGRAM TREATING TIME-DEPENDENT NEUTRON AND PHOTON TRANSPORT THROUGH MATTER

E. S. Troubetzkoy

September 1966

Work Performed on UNC Project 2336
under New York Procurement District
Contract No. DA-30-069-AMC-785(X) for
The United States Army,
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

UNITED NUCLEAR CORPORATION
Research and Engineering Center
Elmsford, New York

INTRODUCTION

UNC-SAM-2, the second United Nuclear Corporation Stochastic Approximation Method, is a code system in the FORTRAN language for evaluating the time dependence of the transport of neutrons or photons through matter using Monte Carlo techniques.

This program represents the latest result of a continuing effort to develop Monte Carlo programs which will handle complex geometrical configurations with improved versatility and speed. It is based upon its predecessor codes UNC-SAM[1] and ADONIS.[2] We are indebted to M. H. Kalos for the development of many of the numerical techniques employed in these predecessor programs and for contributions to some of the improvements which have been incorporated in UNC-SAM-2, and to W. Guber for major contributions to the formulation of UNC-SAM-2 as well as for most of the actual coding involved.

Chapter 1 of this report describes some of the new importance sampling and variance reduction techniques which have been incorporated into the code.

Chapter 2 provides a short description of the UNC-SAM-2 program and each of its major components.

The remaining chapters provide more detailed descriptions of the components of the code with representative input descriptions.

TABLE OF CONTENTS

FIGURES

1. METHODS OF ANALYSIS

1.1 THE BOLTZMANN EQUATION

1.1.1 General

Let us define the density of particles coming out of collision $\psi(P,Q)$, where P represents the spatial coordinates $\vec{X}$, and Q the direction $\vec{\Omega}$ and the energy E.

$\psi(P,Q)$ satisfies the time-independent Boltzmann equation

$$\psi(P,Q) = S(P,Q) + \int \psi(P',Q')\, T(P' \rightarrow P \mid Q')\, C(Q' \rightarrow Q \mid P)\, d\tau_{P'}\, d\tau_{Q'} \qquad (1)$$

where $S(P,Q)$ is the density of source particles, $T(P' \rightarrow P \mid Q')$ is the transport kernel which, per unit particle coming out of collision or born at (P',Q'), gives the density of particles coming into collision at (P,Q').

T has the functional form[3]

$$T[\vec{X} \rightarrow \vec{X} + S\vec{\Omega}'/(E,\vec{\Omega})] = \mu(\vec{X} + S\vec{\Omega}', E)\, e^{-\int_0^S \mu(\vec{X} + S'\vec{\Omega}', E)\, dS'} \cdot \qquad (2)$$

$$\cdot\, \delta(\vec{\Omega}' \cdot \vec{\Omega}_1)\, \delta(\vec{\Omega}' \cdot \Omega_2)\, \eta(\vec{\Omega}' \cdot \vec{\Omega})$$

where $\vec{\Omega}_1$ and $\vec{\Omega}_2$ form an orthonormal set with $\vec{\Omega}$, and

$\eta(X) = 1$ for $X \geq 0$

$\eta(X) = 0$ for $X < 0$

$\mu(X,E)$ = the macroscopic total cross section at energy E of the material in the neighborhood of the point $\overline{X}$.

$C(Q' \rightarrow Q|P)$ is the collision kernel which, per unit particle coming into collision at (P,Q'), gives the density of particles coming out of collision at (P,Q). Details of possible forms of the collision kernel handled by the code are given elsewhere. Here we only assume that it is positive, definite, and integrable with respect to Q.

The flux $\varphi(P,Q)$ is related to the solution $\psi(P,Q)$ of Eq. 1 through

$$\varphi(P,Q) = \int d\tau_{P'}\, \psi(P',Q)\, T(P' \rightarrow P|Q)/\mu(P,Q) \tag{3}$$

1.1.2 The Estimates

The quantities to be computed are of the form:

$$H = \int_V \varphi(P,Q)\, d\tau_P\, d\tau_Q \tag{4}$$

where V is a given region of phase space or, more generally

$$H = \int_V \varphi(P,Q)\, R(P,Q)\, d\tau_P\, d\tau_Q \tag{5}$$

where R(P,Q) is a response function.

We assume here that R(P,Q) is definite and nonsingular, that the integral (Eq. 3) exists, and that the region V is nondegenerate. A degeneration of a specific type [V(P,Q) degenerating to a single point P] is discussed elsewhere.

Substituting Eq. 3 into Eq. 4 we can write:

$$H = \int d\tau_{P'} \int_V d\tau_P\, d\tau_Q\, \varphi(P',Q)\, T(P' \rightarrow P|Q)/\mu(P,Q) \tag{6}$$

1.1.3 Time Dependence

We are concerned with the solution of the time independent Boltzmann equation. Even so, one is sometimes interested in the time dependence of the solution.

The source $S(P,Q,t)$ is assumed to be a known function of time. The velocity of particle is a known function of energy:

$$v(E) = 1.3833 \times 10^6 \sqrt{E} \text{ cm/sec} \qquad (E \text{ in eV}) \text{ for neutrons}$$

$$v(E) = 2.9978 \times 10^{10} \text{ cm/sec} \qquad \text{for } \gamma\text{-rays.}$$

Collisions are assumed to take place instantaneously. Under these assumptions the Boltzmann equation becomes:

$$\psi(P,Q,t) = S(P,Q,t) + \int \psi\left[P',Q',t - \frac{|P-P'|}{v(E)} \right] T(P' \rightarrow P \,|\, Q)\, C(Q' \rightarrow Q \,|\, P) \cdot$$

$$\cdot\, d\tau_{P'}\, d\tau_{Q'}$$

The expression of the flux is:

$$\varphi(P,Q,t) = \int d\tau_{P'}\, \psi\left[P',Q,t - \frac{|P-P'|}{v(E)} \right] T(P' \rightarrow P \,|\, Q)/\mu(P,Q)$$

and the estimator can be generalized to:

$$H = \int_V \varphi(P,Q,t)\, d\tau_P\, d\tau_Q\, dt$$

where V is now a region of phase space including time.

$$H = \int d\tau_{P'} \int_V d\tau_P\, d\tau_Q\, dt\, \psi\left[P',Q,t - \frac{|P-P'|}{v(E)} \right] T(P' \rightarrow P \,|\, Q)/\mu(P,Q)$$

1.2 THE MODIFIED BOLTZMANN EQUATION

1.2.1 General

Let us introduce an importance function $I(P,Q)$, or a weight function $W(P,Q)$.

$$I(P,Q) = \frac{1}{W(P,Q)} \quad .$$

and let us multiply Eq. 1 by this importance function

$$\frac{\psi(P,Q)}{W(P,Q)} = \frac{S(P,Q)}{W(P,Q)} + \int \frac{\psi(P',Q')}{W(P',Q')} \frac{W(P',Q')}{W(P,Q')} T(P' \to P \mid Q') \frac{W(P,Q')}{W(P,Q)} C(Q' \to Q \mid P) \cdot \tag{6a}$$

$$\cdot d\tau_{P'} \, d\tau_{Q'}$$

Let us define a modified density of particles coming out of collision:

$$\tilde{\psi}(P,Q) = \frac{\psi(P,Q)}{W(P,Q)} \quad , \tag{7}$$

a modified source density function:

$$\tilde{S}(P,Q) = \frac{S(P,Q)}{W(P,Q)} \, , \tag{8}$$

a modified transport kernel:

$$\tilde{T}(P' \to P \mid Q') = \frac{W(P',Q')}{W(P,Q')} \, T(P' \to P \mid Q \,), \tag{9}$$

and a modified collision kernel:

$$\tilde{C}(Q' \to Q \mid P) = \frac{W(P,Q')}{W(P,Q)} \, C(Q' \to Q \mid P). \tag{10}$$

Substituting these quantities into Eq. 6a, we obtain the modified Boltzmann equation:

$$\tilde{\psi}(P,Q) = \tilde{S}(P,Q) + \int \tilde{\psi}(P',Q')\, \tilde{T}(P' \rightarrow P \mid Q')\, \tilde{C}(Q' \rightarrow Q \mid P)\, d\tau_{P'}\, d\tau_{Q'} \qquad (11)$$

Substituting them into Eq. 6, we obtain a new expression for H:

$$H = \int d\tau_{P'} \int_V d\tau_P\, d\tau_Q\, \tilde{\psi}(P',Q)\, \tilde{T}(P' \rightarrow P \mid Q)\, W(P,Q)/\mu(P,Q) \qquad (12)$$

1.2.2 The Importance Function

To evaluate the desired quantity, H, one can either solve Eq. 1 and perform the integration (Eq. 5), or solve Eq. 11 and perform the integration (Eq. 12).

As the method of solution is the Monte Carlo method, the estimate of H will be associated with a certain variance. The introduction of an importance (or weight) function gives the freedom to reduce that variance. In fact, it is known that, if the importance function is taken to be the solution of the problem adjoint to Eqs. 1 and 5, the variance associated with the estimate (Eq. 12) vanishes. An approximation to this "best" importance function will lead to finite (but small) variance, and should, if the approximation is good, lead to a variance smaller than the one associated with the unbiased problem.

In the code, we provide the possibility to take into account, in a somewhat crude manner, the spatial, energy, and angular dependence of the weight function.

We assume that the weight is a separable function of energy and angle:

$$W(\vec{X},E,\vec{\Omega}) = W_X(\vec{X})\, W_E(\vec{X},E)\, W_\Omega(\vec{X},\vec{\Omega})$$

where the first term is not really necessary but is introduced because of computer coding reasons.

The computer program is restricted to assemblies consisting of spatial regions, each containing a uniform distribution of materials. The restrictions for the weight function that the code can handle are as follows: $W_X(\vec{X})$ is constant for

all $\vec{X}$ in the same region; $W_E(\vec{X},E)$ has the same energy dependence for all $\vec{X}$ in the same region. The energy dependence is specified as follows. A set of energy bins is specified by the energy boundaries. This set is the same for all regions, but can be different from any other energy bins (like flux bins). For each spatial region, a set of numbers is specified, corresponding to constant energy weight factors in each energy bin. Similarly, the angular weight factor $W_\Omega(\vec{X},\vec{\Omega})$ has the same angular dependence for all $\vec{X}$ in the same spatial region. The angular weight factor is assumed to depend only on $\omega = \vec{\Omega}\cdot\vec{\Omega}_i$, where Ω_i is specified in each region:

$$W_\Omega(\vec{X},\vec{\Omega}) = W_\omega(\vec{X},\vec{\Omega}\cdot\Omega_i) .$$

A set of angular bins is specified, and assumed to be the same for all regions. For each spatial region, a set of numbers is specified, corresponding to constant angular weight factors in each angular bin. There are three different ways to specify $\vec{\Omega}_i$ in each region. The simplest is to specify $\vec{\Omega}_i$ itself as a constant vector (aiming vector) or to specify a point $\vec{R}_i$ (either target or source) with $\vec{\Omega}_i$ specified, at each point $\vec{X}$ in the given region, by

$$\vec{\Omega}_i = \frac{\vec{X} - \vec{R}_i}{|\vec{X} - \vec{R}_i|}$$

or finally to specify a constant angle θ_i with the z-axis, with $\vec{\Omega}_i$ specified, at each point $\vec{X}$ in the given region, by

$$\Omega_{ix} = \frac{x}{\sqrt{x^2+y^2}} \sin\theta_i$$

$$\Omega_{iy} = \frac{y}{\sqrt{x^2+y^2}} \sin\theta_i$$

$$\Omega_{iz} = \cos\theta_i .$$

At present, only the first option is operational in the code.

1.3 SOLUTION OF THE PROBLEM BY THE MONTE CARLO METHOD

1.3.1 General

The numerical estimate of the desired quantity H is obtained as follows: Eq. 11 is solved by Monte Carlo, by producing a population of points (P,Q,t) having a density proportional to $\tilde{\psi}(P,Q,t)$. The estimation of H is then carried out by performing the integration (Eq. 12).

The solution by Monte Carlo goes as follows: a point (P,Q,t) is generated with a probability distribution function proportional to $\tilde{S}(P,Q,t)$. Points where particles enter into collision are generated with a density $\tilde{T}(P_{old} \rightarrow P_{new}|Q)$. The time of flight is calculated. Particles getting out of collision are generated with a density $\tilde{C}(Q_{old} \rightarrow Q_{new}|P)$. The process, picking from $\tilde{T}$ and then from $\tilde{C}$, is repeated until no further possibility exists as explained below. The process is then repeated for another source point, until the desired number of source particles has been exhausted.

1.3.2 Picking from the Modified Source Distribution Function

The code can handle either a source tape generated by an external source generator, or accept as input a description of the source under some restrictions, and generate, internally, particles from that source modified by the importance weights.

The source tape consists of a collection of points (P,Q,t) with a specification of a region number in which they occur, and a set of two weights: W_p and W_c. W_p is the statistical weight of the source point. W_c is a normalization factor carried by the particle and by the entire history, with all scores multiplied by that factor. It is normally set as one (1). Upon picking a particle, the actual weight, W, for the particle (P,Q) is computed as explained in Section 1.2.2, and the factor

$F = W_p/W$ is calculated and interpreted as being the number (which may be fractional) of independent source particles which have coordinates (P,Q,t).

The built-in source generator accepts a function S(P,Q,t) with the following restrictions:

$$S(\vec{X},E,\vec{\Omega},t) = S_X(\vec{X})\, S_E(E)\, S_\Omega(\vec{\Omega})\, S_t(t).$$

$S_X(\vec{X})$ is assumed to be constant for all $\vec{X}$ in the same region; power densities must be specified for all source regions.

$S_E(E)$ is specified by a table of

$$\int_E^\infty S_E(E)\, dE \text{ vs } E.$$

Linear interpolation is assumed on

$$\ln \int_E^\infty S(E)\, dE \text{ vs } E.$$

A table representing the Cranberg fission spectrum[4] is built into the code, but different tables can be read in as an option. E_{high} and E_{low}, high and low energy cutoffs of the problem must also be provided. At present, $S_\Omega(\vec{\Omega})$ can be either isotropic or monodirectional only (in the latter case, the direction of the source must be specified). $S_t(t)$ is specified by a table of

$$\int_0^t S_t(t)\, dt \text{ vs } t.$$

Linear interpolation is assumed between entries in the table.

Once the function S(P,Q,t) has been specified, the code will pick source particles directly from a probability distribution function proportional to the modified source density function $\tilde{S}$.

$$\tilde{S}(P,Q,t) = \frac{S(P,Q,t)}{W(P,Q)} .$$

The mechanics of the picking routine are described elsewhere. Here we want only to mention that points P,Q,t are generated when needed. $P,\vec{\Omega}$, and t are picked at random from $\tilde{S}$, whereas the energy E is picked in a stratified way, in descending order for each statistical aggregate.

As we pick from a renormalized modified source distribution, answers (fluxes) have to be renormalized. They can be normalized either to a unit source particle, or to a total power. The normalization is achieved by multiplying all the region weights $W_X(\vec{X})$ by a constant normalization factor, once and for all, at the beginning of the problem.

The internal picking routine delivers a point (P,Q,t) with a specification of the region number in which it occurs, and a set of two weights: W_p, equal to the actual weight W for the particle (P,Q); and $W_c = 1$. The factor $F = W_p/W$ is then equal to unity, and the particle is a bona fide single particle with coordinates (P,Q,t).

1.3.3 Picking from the Modified Transport Kernel and Scoring of Fluxes

Once a particle or, more generally, F particles (P,Q,t) are picked from the modified source distribution function (or from a modified collision kernel), points where particles (P,Q) enter into collision must be generated from a distribution which is F times the modified transport kernel.

Looking at Eq. 2 of Section 1.1, and at Eq. 9 of Section 1.2, the points P are defined by $\vec{P} = \vec{P}' + S\vec{\Omega}$, where S has the distribution function F(s):

$$F(s)\,ds = \frac{FW(\vec{X}',E\vec{\Omega})}{W(X'+s\vec{\Omega},E\vec{\Omega})}\,\mu(\vec{X}'+s\vec{\Omega},E)$$

$$\exp f\left[-\int_0^s \mu(\vec{X}'+s'\vec{\Omega},E)\,ds'\right]ds$$

The distribution F(s)ds is not, in general, a probability distribution function, as it is not necessarily normalized to unity. The original transport kernel, as well as

$$\mu \exp\left(-\int_0^s \mu ds\right)$$

are normalized to unity. But the biasing function

$$\frac{W(X',E\vec{\Omega})}{W(X'+s\vec{\Omega},E\vec{\Omega})}$$

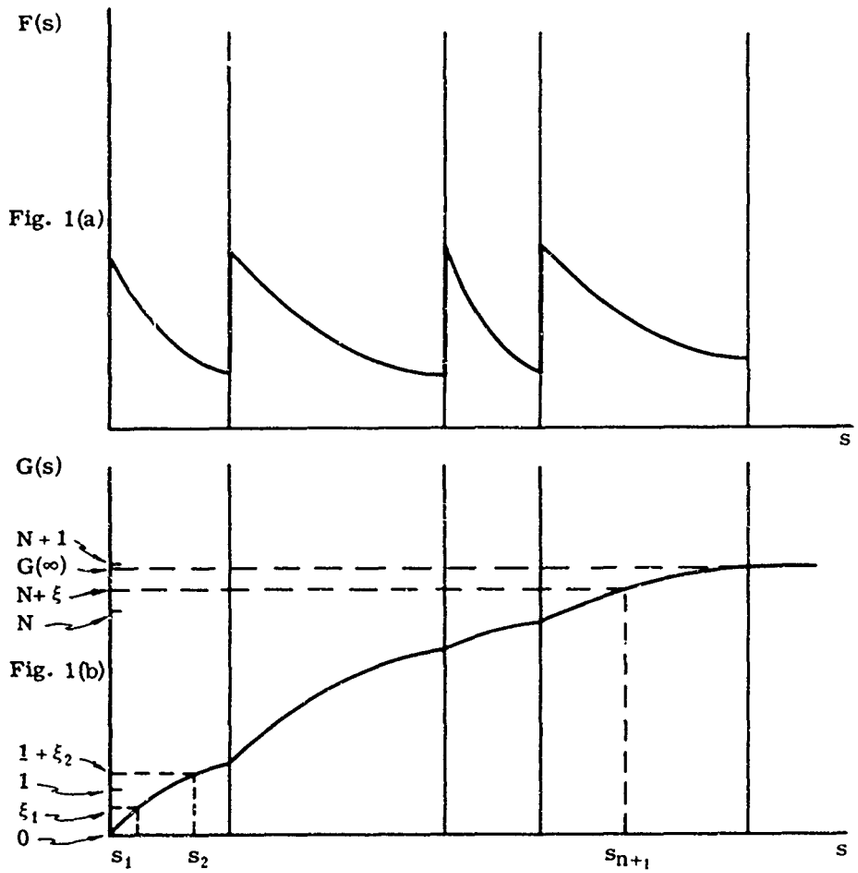
in general destroys that normalization. In addition the factor F is sometimes different from unity.

Fig. 1(a) shows a schematic plot of the function F(s), drawn under the assumption that $\vec{\Omega}$ points into a direction of increasing importance (i.e., of decreasing weights). As can be seen, the s-dependence is $e^{-\mu s}$ in each region, with a discontinuity at each boundary where a change in weight and (or) of μ occurs.

To pick s from such a distribution, we construct the function:

$$G(s) = \int_0^s F(s)\,ds$$

G(∞) gives the normalization of F(s).



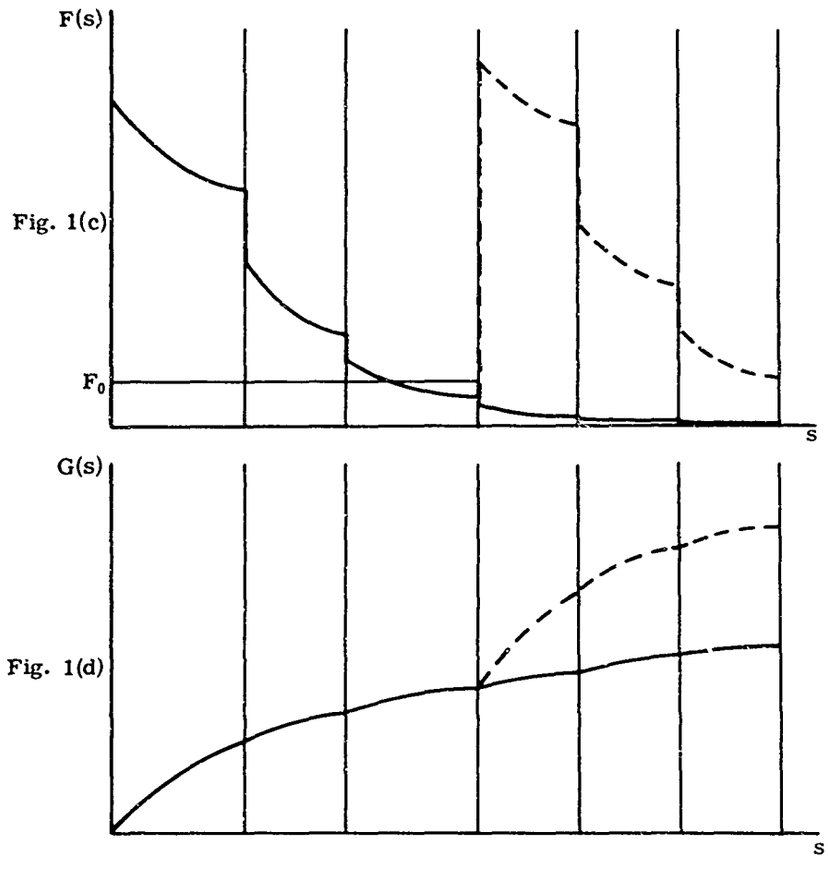


Fig. 1 — Schematic Diagram of the s-Dependence of the Modified Transport Kernel F(s) and of the Integral

$$G(s) = \int_0^s F(s)\, ds$$

If N is an integer such that $N \leq G(\infty) < N + 1$, we will arrange to pick either N or N + 1 distances s, with probability $[N+1-G(\infty)]$ or $[G(\infty)-N]$, respectively. Furthermore, the collision points will be picked in a stratified way.

To pick the first collision point, we pick a random number ξ_1 and solve [see Fig. 1(b)]

$$G(s_1) = \xi_1.$$

To pick the second collision point, we pick another random number ξ_2 and solve

$$G(s_2) = 1 + \xi_2, \text{ etc.}$$

Before solving for each s_n, $n=1, \ldots, n+1$, $G(s_n) = n-1+\xi_n$, we test whether such a solution exists, i.e., whether $G(\infty) > n-1+\xi_n$. If the test is not passed, no n^{th} and no further collisions should be generated from that flight.

Actually, we do not pick n different random numbers, but just a single one at the beginning of the flight, and set $\xi_2 = \xi_3 = \ldots. = \xi_n = \xi_1$.

In the actual coding, the tracking, integration, and selection of collision points is done simultaneously.

In describing the process, it is implied that we always have to track to ∞ (or to "escape") in order to distribute collision points. To avoid unnecessary tracking into regions of decreasing importance, we test, at each boundary, the integrand F(s) vs a fixed cutoff value F_0 (a number less than 1, which we like to set as 0.05). If, at some s_0, $F(s_0)$ becomes smaller than F_0, we Russian-roulette the remaining flight; with probability $[1-F(s_0)]$ we cut off the tracking, whereas with probability $F(s_0)$ we multiply the remaining density function F(s) by $1/F(s_0)$ and continue tracking. The case is illustrated in Figs. 1(c) and 1(d).

The time, t, at which the interaction is entered at distance, s, is given by $t = t' + s/v(E)$.

Fluxes are being scored simultaneously with tracking. As seen from Eq. 12 of Section 1.2, we have to perform the integral

$$H = \int d\tau_{P'} \int_V d\tau_P \, d\tau_Q \, \tilde{\psi}(P',Q) \, \tilde{T}(P' \rightarrow P \mid Q) \, W(P,Q)/\mu(P,Q)$$

or, if we are interested in time dependence, the integral

$$H = \int d\tau_{P'} \, dt' \int_V dt \, d\tau_P \, d\tau_Q \, \tilde{\psi}(P',Q,t') \, \delta\left[(t-t') - \frac{|P-P'|}{v(E)}\right] \tilde{T}(P' \rightarrow P \mid Q) \, \frac{W(P,Q)}{\mu(P,Q)} .$$

Let us assume that

$$V = V_P V_E V_t$$

where V_P = a certain region of space

V_E = a certain energy interval

V_t = a certain time interval,

the expression of H becomes

$$H = \int d\tau_{P'} \int dt' \int d\tau_\Omega \int_{V_E} dE \, \frac{1}{F} \, \tilde{\psi}(P',E,\vec{\Omega},t') \int_{V_t} dt \int_{V_P} d\tau_P \, \delta\left[(t-t') - \frac{|P-P'|}{v(E)}\right] \cdot$$

$$\cdot F\tilde{T}(P' \rightarrow P \mid E,\vec{\Omega}) \cdot \frac{W(P,E,\vec{\Omega})}{\mu(P,E)}$$

The first four integrations are performed by Monte Carlo in the process of picking from the source distribution and collision kernels. The last two integrations are performed analytically.

$$\int_{V_t}\int_{V_P} d\tau_P\ \delta\left[(t-t') - \frac{|P-P'|}{v(E)}\right] F\tilde{T}(P'-P\,|E,\vec{\Omega})\, W(P,E,\vec{\Omega})/\mu(P,E) =$$

$$= \int_{V_t} dt \int_{V_s} ds\ \delta\left[(t-t') - \frac{s}{v(E)}\right] F(s)\, W(\vec{X}'+s\vec{\Omega},E,\vec{\Omega})/\mu(\vec{X}'+s\vec{\Omega},E)$$

as they involve the same integrations needed to obtain the cumulative distribution G(s).

1.3.4 Picking from the Modified Collision Kernel

Given a particle (P,Q',t) going into collision, we are supposed to pick particles (P,Q,t) from the modified collision kernel:

$$\tilde{C}(Q'\to Q\,|P) = \frac{W(P,Q')}{W(P,Q)}\ C(Q'\to Q\,|P)$$

Picking is not done, as it should be, directly from $\tilde{C}$, but rather from the original collision kernel. However, the weight W(P,Q') is attached to the particle. Upon the particle being picked up, the fraction

$$F = \frac{W(P,Q')}{W(P,Q)}$$

is calculated, and interpreted as being the number (which may be fractional) of independent particles coming out of collision with coordinates (P,Q,t). These will be split or Russian-rouletted during transport as we have seen in the preceding section.

The possible forms of the collision kernel, and methods to pick from it, are described in UNC-5093.[1] The present code uses the same methods.

2. SHORT DESCRIPTION OF THE CODE

The code system is written in FORTRAN for the Control Data Corporation's (CDC) 1604-A computer and for the Ballistic Research Laboratory's BRLESC computer. A 32-K core and from 2 to 8 magnetic tapes are required. UNC-SAM-2 is, in reality, a chained series of independent programs which process cross-section and geometry data, do the transport problem, and edit the results. A brief summary of the independent programs contained in UNC-SAM-2 follows. Fig. 2 indicates the information flow in the system.

GENDA

The Generation of Data routine (GENDA) starts with a fundamental set of neutron or photon cross-section data tabulated as a function of energy. It allows for the basic tabulation of the total cross section, the scattering cross section, the Legendre expansion coefficients of the differential scattering cross section, the inelastic scattering cross section for both level and continuum scattering, etc. These data are tabulated generally on differing energy meshes. GENDA processes the data to generate a set of cross-section data on a desired cross-section set by either linear or logarithmic interpolation. The spectrum of neutrons emitted by continuum inelastic scattering, according to statistical theory, is used for heavier materials. For lighter materials, such as Li and Be, special treatments are available. The output of this routine is a tape which may be used as input to GENPRO.

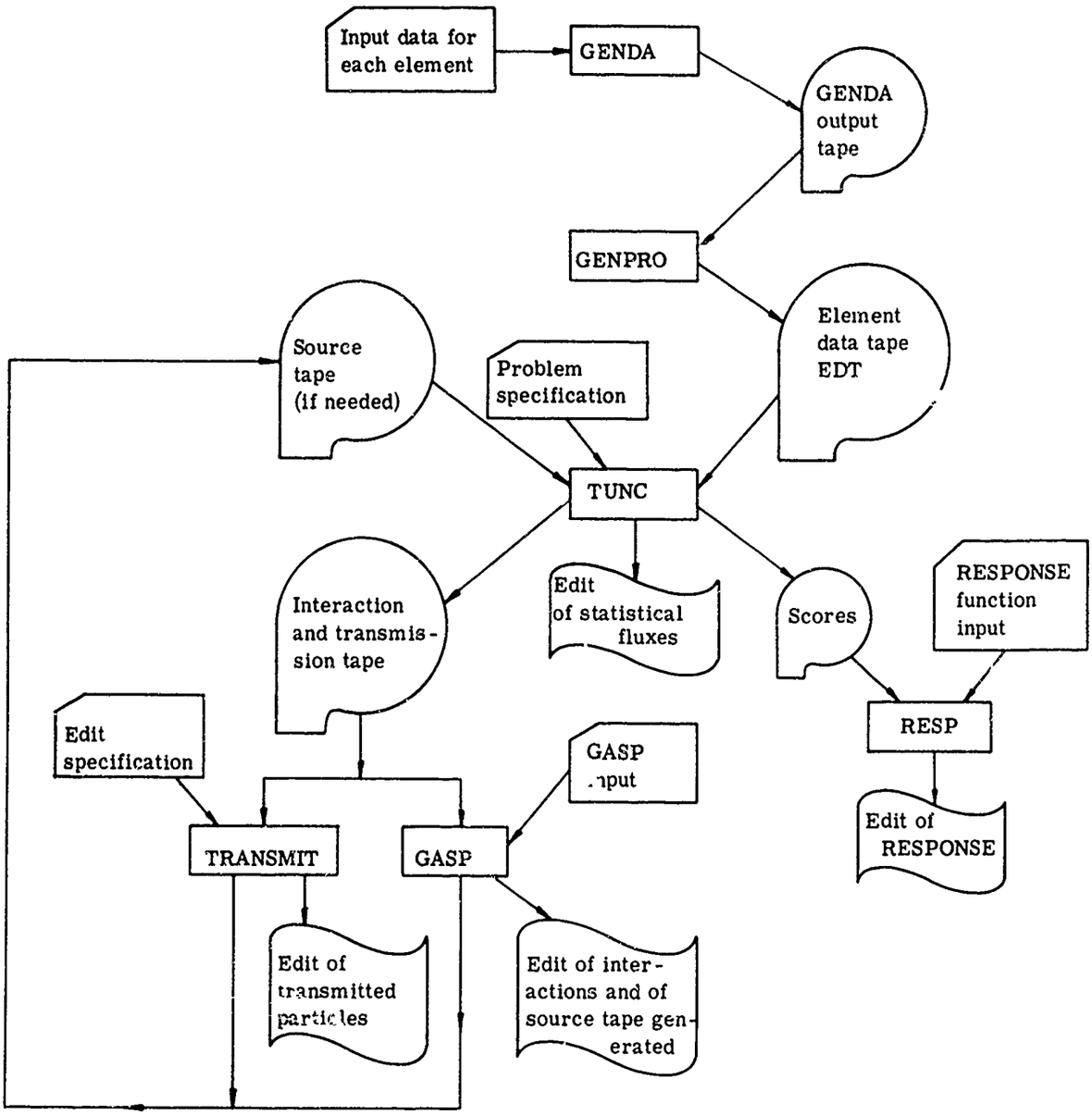


Fig. 2 — SAM-2 Flow Diagram

GENPRO

The GENDA Processor (GENPRO) routine uses the GENDA output to determine the tables of probabilities peculiar to a nuclide, required for a Monte Carlo code, at a prescribed energy mesh. For instance, the probability that a neutron suffers an elastic scattering is given by the ratio of elastic-to-total-scattering at that energy. The probability of inelastic scattering from one energy to another is computed in this routine. At its conclusion, one has a library tape, called the Element Data Tape (EDT), on a given energy mesh to be used in subsequent processing.

Up to this point, no problem-oriented input is needed. The functioning of later routines (TUNC, GASP) will be dependent on such problem-oriented data.

Both GENDA and GENPRO codes are described in detail in UNC-5093.[1]

TUNC

TUNC is the main program of an overlay tape. Fig. 3 indicates the information flow in the system. It essentially calls in sequence the following subprograms.

BAND

Given the concentrations of the materials contained in the configuration, an Organized Data Tape (ODT), bearing the total macroscopic cross sections as well as other probabilities for mixtures of nuclides, is created by the BAND routine. These are the problem-dependent cross-section data tabulated against energy at the prescribed energy mesh previously used in GENDA and GENPRO.

BEDIT

Edits an existing Organized Data Tape.

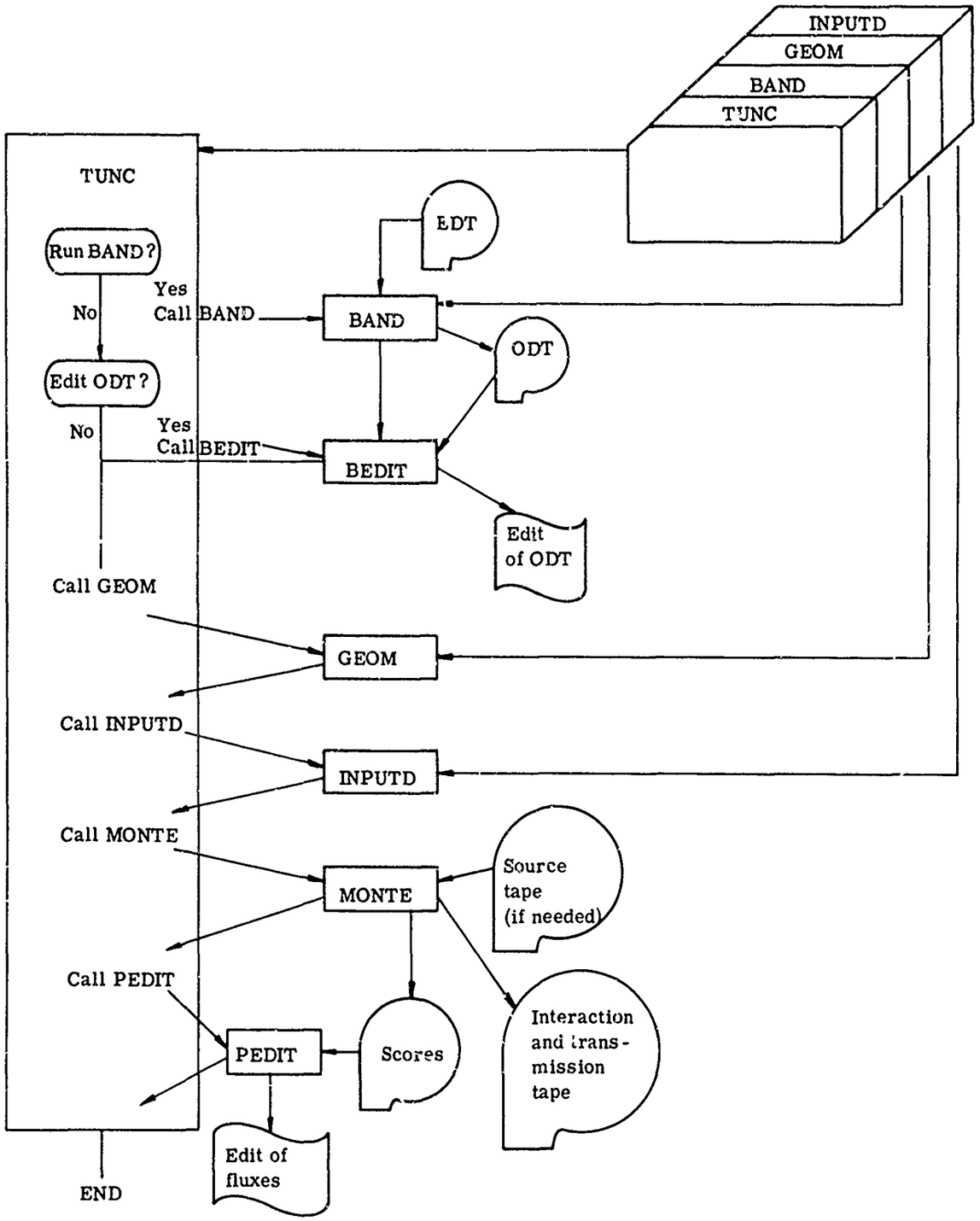


Fig. 3 — TUNC Flow Diagram

GEOM

This is a routine which takes a simplified geometrical description of the physical system, as provided by the problem originator, and produces the rather complex set of data required by the transport program.

Three types of geometries are available: Type 1 is for axially symmetrical problems; Type 2 is for spherically symmetrical problems; and Type 3 is for more complicated geometries, in which case the problem originator must decompose space into a set of boxes known as "ordinary regions." Inside these boxes may be placed "nonordinary regions" consisting of spheres, cylinders, wedges, or other boxes. Nonordinary regions may enclose other nonordinary regions.

INPUTD

Reads in the remaining problem specifications, namely, the assignment of compositions to geometrical regions, the assignment of regions, energy, and angular dependent statistical weights, and the assignment of scoring regions (i.e., regions where average fluxes are to be calculated), as well as the positions of point detectors.

The source specification is also read in. The source can be defined by either an external source tape, or by a description of the energy spectrum and by a power density in different regions.

MONTE

Given the output of the previous subprograms, one is now ready to proceed with the transport program MONTE. The program picks source particles, tracks them to escape (scoring flux contributions where needed), distributing (and storing) collision points. Particles coming out of collision (if any) are subject to the same treatment as source particles. The calculation is done

in statistical groups so that variances can be calculated. The energy range is broken up into supergroups to decrease memory limitations. Information (cross sections and scores) relevant to only one supergroup is in the computer memory at any time. Answers for each statistical group and each supergroup are written on a scratch tape.

PEDIT

Reads and edits the output tape of MONTE, giving fluxes and standard deviations.

TRANSMIT

TRANSMIT accepts as input an interaction and transmission tape generated by MONTE. It selects only transmitted particles, i.e., particles which enter a given region during the MONTE tracking, and writes out a source tape consisting of these particles. It also edits that tape.

GASP

The routine Gamma Source Particle generator (GASP) computes gamma-ray production from a primary neutron problem to serve as a source for a secondary gamma-ray problem. GASP requires:

1. An interaction tape (generated by a previous TUNC problem) containing a record of all interactions able to cause gamma-ray emission
2. Probability numbers for each nuclide of the problem giving the probable number of gamma rays emitted as a function of energy.

The routine can cope with the generation of other secondaries upon modification of the probability number input. The output of either GASP or TRANSMIT is a source particle tape which becomes input to TUNC.

RESPONSE

This program accepts as input the output tape of MONTE. It calculates integrals $\int R(E)\,\varphi(E)\,dE$ and associated statistical errors for specified functions R(E) in specified regions.

3. DATA GENERATOR AND PROCESSOR ROUTINES (GENDA AND GENPRO) AND THE MAIN CODE, TUNC

The GENDA and GENPRO routines are described in detail in UNC-5093.[1] These codes have not been modified. Nuclides are identified on input to GENDA by a 10-digit integer and by an atomic mass. In our old Monte Carlo codes, the only relevant identification of the element was the integral part of the atomic mass. In the present code, however, the relevant identification is the integer consisting of the five right-most digits of the 10-digit number read in by GENDA. A convenient way to identify an element is ZZAAA where ZZ is the atomic number and AAA the mass of the isotope (set to zero for the natural element). It is mentioned elsewhere in the UNC-5093 report[1] that (n,2n) reactions are ignored in the interaction routines, except for beryllium for which two neutrons are emitted following a reaction identified as inelastic. This is still true in the present code, beryllium being recognized as having an identifier 04AAA (any AAA).

TUNC is the main program of an overlay tape. The input consists of two cards, a title card and a data card containing IBAND and NBAND.

If IBAND = 0	TUNC will call BAND which creates an Organized Data Tape (ODT), and then will call BEDIT which edits the ODT
If IBAND = 1	TUNC will call BEDIT, which edits an existing ODT
If IBAND = 2	TUNC bypasses both BAND and BEDIT, and programs GEOM, INPUTD, MONTE, and PEDIT are called sequentially.

4. BAND

The purpose of BAND is to arrange the cross-section data in an organized fashion for the particular problem in process.

The GENDA and GENPRO programs are used to generate an Element Data Tape (EDT) which contains microscopic cross sections and probability tables for all nuclides of interest. This EDT can be considered as a library tape for any subsequent Monte Carlo problems.

For a specific problem, however, the system is divided into regions which contain given nuclide compositions. Each composition may be described by a set of nuclides and their respective concentrations. When a particle enters collision at some point within a region of given composition, the Monte Carlo program has to select the nuclide in the composition with which the particle collides.

The probability that a particle with energy E will interact with nuclide k of the composition R is given by

$$P(k,R) = \frac{C_{k,R}\sigma_T(k,E)}{\mu_{T,R}(E)}$$

where $\mu_{T,R}(E)$ is the total macroscopic cross section of the composition, and $C_{k,R}$ is the concentration of nuclide k in region R.

A BAND input form is shown in Chapter 13. The BAND output is a long array in variable dimension, where floating cross-section numbers are packed together with fixed number indices in single words. Chapter 5 describes the edit of the ODT.

5. BEDIT

BEDIT edits the ODT generated by BAND. The edit is by energy band.

First Line

BAND-FLAG, sequential number of the band being edited.

Second Line

NO. of BANDS, total number of bands in problem.

Third Line

Is a heading for the table that follows.

JEN1 is the location of the first word of the energy mesh table;

JEN2 is the location of the next to the last word of the energy mesh table.

Next Heading

REGION	Composition number
LOCMU	The location of the first word of the macroscopic total cross-section table for that composition
LOCCON	The location of the first word of the table giving the concentrations of different nuclides.

This is followed by a table for all compositions present in the problem.

Next Heading

CONC	Concentration of nuclide
LOC	Location of first word of microscopic cross-section data for the nuclide
ATWT	Atomic weight of the nuclide.

This is followed by a table for all nuclides in each composition present in the problem.

This is followed by a printout for each nuclide present in the problem.

Heading

LOELEV	Location of first word of table giving discrete energy levels
ATWT	Atomic weight of nuclide
LPLEV	Number of discrete energy levels
LCHI	Length of CHI-table
LENN	Length of ENN-table

The meaning of CHI and ENN tables is described in the GENPRO section of UNC-5093. We will mention here that the angular distribution (E, cos θ) is divided into N equally probable intervals of cos θ. The N+1 boundaries of cos θ define N+1 values of $\chi = (1 + \cos \theta)/2$, which are stored in a CHI table for the energy E. The length of the CHI-table is N+1.

Similarly the spectrum of inelastic neutrons P(E,E′) is divided into M equally probable intervals of E′, and the M+1 boundaries of E′ are stored in an ENN-table for the energy E. M+1 is the length of the ENN table.

Next Heading

E	Total cross section, in barns
SIGMA	If CHI = 1, elastic scattering is isotropic in the CM system. If = 2, elastic scattering is with hydrogen (special routine).

SIGMA (Continued)	If = 4, elastic scattering isotropic in the lab system. If = 5, the Klein-Nishina formula is to be used (γ-rays only). If > 5, then CHI is the location of the first word of the appropriate CHI-table.
SCATTER	Elastic scattering cross section, in barns
PLEV	If = 0, inelastic scattering is not via discrete levels. If > 0, PLEV is the location of the first word of a table giving probabilities to excite the various levels.
ABSORB	Absorption cross section, in barns
ENN	If = 0, inelastic scattering does not give rise to a continuous spectrum of inelastic neutrons. If > 0, ENN is the location of the first word of the appropriate ENN table
L	The location of the total cross-section entry.

This is followed by a listing of all the tables referred to (CHI, ENN, PLEV, ELEV), together with their relative locations.

After the printout for all the elements comes the printout of macroscopic cross sections for all the compositions, in cm^{-1} vs E, in eV.

This is the end of the edit for neutron problems. For γ-rays, one has, in addition, a printout of $\Sigma_i C_i Z_i$ for each composition, where C_i is a concentration and Z_i is an atomic number.

6. GEOM

Monte Carlo programs track simulated particles through a specified geometrical configuration, undergoing all the interactions that an actual particle is expected to undergo. The particle flight may pass through media having different properties. Therefore, it is of primary importance to know at all times (1) in which medium the particle is, (2) when the particle leaves the medium, and (3) which medium it will be entering. Different media may be described as containing given material compositions in three-dimensional geometrical figures.

However, it is necessary to decompose these figures into particular geometrical forms which can be handled by the MONTE program.

The GEOM program processes the input which consists of the description of the desired configuration by means of these elementary geometrical forms and stores the information into tables for rapid data access in subsequent Monte Carlo programs.

Three distinct types of geometries can be handled:

1. Axially symmetric geometries (MASQUE)
2. Spherically symmetric geometries
3. Other geometries (UNC-SAM geometry)

The input common to all three types of geometries is a title card (80 characters) followed by a data card giving:

ITYPE	Type of geometry 1 - MASQUE 2 - SPHERICAL 3 - UNC-SAM
NRGN	Number of distinct regions
NSRF	Number of distinct surfaces
IREX	Ordinal number of the region declared to be the escape region (>NRGN).

6.1 AXIALLY SYMMETRIC GEOMETRY (MASQUE)

The axis of symmetry is defined to be the z-axis. The geometry is specified in the (z, positive r) half-plane. Regions in this half-plane are restricted to convex quadrangles. The geometry is specified by first defining all the NSRF bounding lines, and then defining each of the NRGN regions by specifying the four lines which bound them, and by giving information on the neighboring regions, as explained below.

An input form for MASQUE is shown in Chapter 13.

The description of the bounding lines is as follows. First one numbers all the NSRF lines (including the z-axis), from one (1) to NSRF, in arbitrary order. One then defines each of the lines in sequential order, by any two points (R_1, Z_1) (R_2, Z_2) on the line. The input for each line consists of a single card with R_1, Z_1, R_2, Z_2, I, where I is the ordinal number of the line.

All the quadrangular regions must also be numbered from one (1) to NRGN in arbitrary order. One then has to define each of the regions, in sequential order. For each region IR, a card gives

$$L_1, L_2, L_3, L_4, J_1, J_2, J_3, J_4, IR$$

where L_i, i = 1, 2, 3, 4 are the ordinal numbers of the four bounding lines of region IR, given in either clockwise or counterclockwise order. If there is a single

neighbor of region IR across line L_i, then J_i is set equal to the number IR′ of that neighboring region. If there is more than one neighbor, say n_i neighbors across boundary L_i, the boundary L_i is called a complex line and J_i is set equal to $(-n_i)$. If the region IR has no complex lines, the input for that region is complete. If complex lines exist, further input is required for each of the complex lines, in order of increasing i.

To specify the n_i neighbors of region IR across the complex line L_i, one has to specify the n_i region numbers IR'_j and the coordinates of the points along line L_i where neighbors change. Actually, as these points lie on a line which is already defined, only one of the two coordinates is sufficient. The criterion for the choice of the coordinate (r or z) is that, if the absolute value of the slope of L_i in the z vs r system is less than one (1), the r coordinates are to be used. The z coordinates are to be used in the other case. Let us call Q the coordinate used. The input consists of n_i pairs of numbers

$$(IR'_1, Q_1)\ (IR'_2, Q_2) \qquad\qquad (IR'_{n_i}, Q_{n_i})$$

where $Q_1 < Q_2 < \ldots., Q_n$, thus signifying that regions IR and IR'_j touch each other along the segment of line L_i between points Q_{j-1} and Q_j (Q_0 is understood to be the low Q vertex of line L_i).

If a region IR is adjacent to the escape region, the corresponding IR′ should be set equal to the escape region number IREX. Regions that include the z-axis in three dimensions are bound by the z-axis (and three more lines) on the r-z plane. The region adjacent to such region across the z-axis is itself, the corresponding IR′ is therefore equal to IR. The quadrilateral bounding a region is allowed to degenerate to a triangle, but a dummy fourth line, passing through a vertex and not crossing the triangle, must be included in the specification. Any number can be used to specify the region neighboring the triangular region across such a dummy line. Note also that the whole configuration must be convex, completely

surrounded by the escape region, and that each point within the configuration must belong to some region IR $\leq$ NRGN, IR $\neq$ IREX.

6.2 SPHERICALLY SYMMETRIC GEOMETRY

In this case regions are defined as spherical shells. The input consists of NRGN radii in increasing order, which define the NRGN+1 regions; region 1 extends from the center to the first radius. (The input NSRF and IREX to GEOM is irrelevant if ITYPE = 2.) An input form in shown in Chapter 13.

6.3 UNC-SAM GEOMETRY

In the UNC-SAM geometry, all space is divided into nonoverlapping rectangular parallelepipeds designated ordinary regions. Spheres, right-circular cylinders, rectangular parallelepiped or right wedges (nonordinary region) with arbitrary orientation can be wholly contained within an ordinary region or another nonordinary region. By suitable combinations of these regions it is possible to represent many complicated configurations.

This geometry is described in detail in UNC-5093.[1] An input form in shown in Chapter 13.

7. INPUTD

The purpose of this routine is to read and store in organized fashion the remaining problem specifications. An input form is shown in Chapter 13.

The first card contains the limits of the problem as shown below:

NSTART	The ordinal number of the first history to be treated (usually 1)
NSTOP	The ordinal number of the last history to be treated
NSTAT	Number of histories per aggregate for statistical scoring of flux
NRMAX	Number of distinct geometry regions
NG	For a neutron problem set NG = 0; for a gamma problem, NG = 1
NT	Number of output time bins (if NT = 0, time dependence is bypassed)
NOUT	Number of output energy bins
NUMSC	Number of flux scoring regions
NRWL	Number of distinct region weights
IREX	Ordinal number of the escape region
IRT	Ordinal number of the transmission region (coordinates of particles crossing a boundary into that region will be written on magnetic tape as "transmitted particles").

The next cards deal with point detectors:

NDET — Total number of point-detectors

NDFAP — Number of detectors for which finite variance estimator is to be used. Scores for the remaining (NDET-NDFAP) detectors will be by statistical estimation (more details will be given in Section 8.4).

The next NDET cards give the coordinates x, y, and z of the NDFAP, and (NDET-NDFAP) detectors.

The next card gives other limits of the problem:

ECUT — Low-energy limit. Particles degrading below E = ECUT will be scored as "degrades" and their tracking and scoring is terminated. This energy cutoff should be set higher or equal to the low-energy limit of cross-section input.

ETH — Irrelevant in the present version of the code.

TCUT — Remote time cutoff: tracking and scoring of particles aging beyond a time T > TCUT is terminated.

FZ — Minimum allowed value of F (see Chapter 2). (Preferred value FZ = 0.05.)

EHIGH — High-energy cutoff of the problem. It should be set smaller or equal to the high-energy limit of cross-section input.

The next cards give the (NOUT+2) output energy bin boundaries. The first and the last energy boundaries must be, and some (or all) of the intermediate boundaries can be, flagged by minus signs. The energy ranges defined by such flagged boundaries is defined as output supergroups, the significance of which will be explained in Section 8.1. The first energy boundary must be $\geq$ EHIGH. The last one must be $\leq$ ECUT.

The next cards give the (NT+1) output time bin boundaries. The first time boundary must be $\geq$ TCUT. The last one must be equal to 0.

The next cards give the NRWL region weights. These NRWL region weights are in effect numbered from one (1) to NRWL, and later on will be referred to by their ordinal number.

The next NRMAX cards define the parameters of each of the NRMAX geometrical regions, the first card applying to region 1, the second to region 2, etc. Each card gives:

ISC	If ISC > 0, fluxes in this geometrical region will be scored in the scoring region with ordinal number ISC. It is possible to have the same scoring region number assigned to several different geometrical regions. If ISC = 0, fluxes will not be scored.
NREG	Ordinal number of the composition of materials present in the geometrical region described by the card
IRW	Ordinal number of the region weight assigned to the geometrical region described by this card
IEW	Ordinal number of the energy weight set assigned to the region (if 0 – no energy importance in the region)
IAN	Ordinal number of aiming vector assigned to the region (if 0 – no angular importance in the region)
IANG	Ordinal number of angular weight set assigned to that region.

After these (NRMAX) cards, the different energy weights must be defined. Several sets of distinct energy weight sets can be given. One defines a single set of energy bins, and each energy weight set is defined by a set of corresponding weights assumed to be constant in an energy bin.

The next card gives:

NEWL	Number of energy bins for energy-importance weights
NEW	Number of distinct energy weight sets.

If these two are nonzero, the next cards give the (NEWL+1) energy boundaries, and the (NEW) weight sets. The directional importance is specified next:

NAIML	Number of distinct aiming vectors
NUMANL	Number of angular bins for angular importance
NUMANG	Number of distinct angular bin sets.

If those three numbers are nonzero, the next cards give the x, y, and z projections of the (NAIML) aiming vectors, the (NUMANL+1) boundaries of cos θ defining the angular bins for energy weighting (the first number must be $\geq 1.$ and the last must be $\leq -1.$), and finally the NUMANG sets, each set consisting of (NUMANL) weights, with each weight assumed to be constant in the corresponding angular bin.

The use of region, energy, and directional weights is explained both in Chapter 1 and Section 8.3. But to clarify the input form, we will recall that the weight of a particle at a given $\vec{X}$ flying with energy E in direction $\vec{\Omega}$ is given by

$$W(X, E, \vec{\Omega}) = W_{\vec{X}}(\vec{X})\ W_E(E)\ W_{\vec{\Omega}}(\vec{\Omega}).$$

Knowing $\vec{X}$ we know the geometrical region number, and therefore, IRW, IEW, IAM, and JANG.

IRW is the ordinal number of the region weight, therefore, $W_{\vec{X}}(\vec{X})$ is known. If IEW = 0, $W_E(E) = 1$; if IEW > 0, one has to find the energy bin where E belongs, and look up the corresponding energy weight $W_E(E)$ from the energy weight set number, IEW. If IAM = 0, $W_{\vec{\Omega}}(\vec{\Omega}) = 1$; if IAM > 0, one looks up the IAMth vector $\vec{\Omega}_{IAM}$, computes cos $\theta = \vec{\Omega}\cdot\vec{\Omega}_i$, locates the angular bin where this cos θ belongs, and looks up the corresponding angular weight $W_{\vec{\Omega}}(\vec{\Omega})$ from the angular weight set number, IANG.

The program INPUTD reads in, prints back, and stores (in packed form) all the information mentioned above, and lays out the memory to provide storage area for the different scores expected in the Monte Carlo part of the calculation. It then calls the subroutine SOUCAL which reads in and processes the source data.

Source Data

An input form is shown in Chapter 13 for SOURCE.

The first card contains:

NSR	Number of regions where the source extends. If this number is 0, an external source tape is expected, and no further input is required.
IFLAG	If IFLAG = 0, a built-in Cranberg fission spectrum will be used for the energy spectrum. If IFLAG > 0, the spectrum is specified later in input. It will be defined by IFLAG entries.
ISW	Switch determining the normalization of problem. If it is 0, fluxes will be normalized to a single (unbiased) source particle. If it is 1, fluxes will be normalized to the "total power" of the source as defined below.

This is followed by (NSR) cards specifying the (NSR) source regions. Each of these cards gives:

ISR	Specification of a geometrical region number
P	Power density in that region
ISO	Flag indicating angular distribution: ISO = 0, isotropic ISO = 1, monodirectional. If ISO = 1 in any of the (NSR) regions, a monodirectional source will be assumed in all the (NSR) regions.

After this follow the (IFLAG) cards specifying the energy spectrum assumed to apply to all the source regions. This is a table of E vs F(E) where

$$F(E) = \int_E^\infty S(E)\,dE.$$

Linear interpolation is assumed on E vs log [F(E)]. The first entry must be for $E \gtrsim E_{high}$, and the last for $E \leqslant E_{cut}$.

If there is time dependence (NT > 0), the time dependence of the source must be specified. A card gives:

NOT Number of cards. This card is followed by NOT cards giving

$$t \text{ vs } \int_0^t S_t(t)\, dt.$$

Linear interpolation is assumed between entries in the table.

Finally, if the source is monodirectional, the projections Ω_x, Ω_y, Ω_z of the direction must be given on the last card.

The subroutine SOUCAL reads in all this input, prints it back, and pre-computes tables to pick directly from the biased source distribution. The code first pre-computes a table of

$$\text{SPEC}(I) = \int_{E_I}^{\infty} S(E)\, dE,$$

where the E_I's take the values of E_{high}, of all the energy boundaries where the energy weight changes, and E_{cut}. The code then runs through all the source regions, and, for each new energy-importance set encountered, pre-computes a table of

$$\widetilde{\text{SPEC}}(I,J) = \int_{E_I}^{\infty} \frac{S(E)\, dE}{W_E(E)}$$

where J runs from 1 to the total number of different energy-importance sets encountered in the source regions; also for each new angular importance set encountered, a table

$$P(I,K) = \int_{\omega_i}^{1} \frac{d\omega}{W_\omega(\omega)}$$

where the ω_i's take the values of cos θ at which the angular weight changes, and K runs from 1 to the total number of different angular importance sets encountered. The different tables are renormalized, and both the modified and unmodified integrated source are computed in each source region. The former quantities are proportional to the probability with which particles should be picked in different source regions. A table SOUR(L) is built up, which gives the cumulative probability for a source to be picked in the M^{th} region for $M \geq L$.

8. MONTE

8.1 THE MAIN PROGRAM

The program MONTE is the main program of an overlay where all the tracking and collision mechanics are performed. The program arranges the calculations in supergroups. We have seen that the program BAND arranges cross sections in certain energy bands. The output energy bins are also arranged in certain output supergroups. Cross-section input corresponding to a single band can be stored in computer memory at any given time. Scores corresponding to a single output supergroup can be stored in computer memory at any given time. The two meshes (bands and output supergroups) do not necessarily coincide. A combined mesh defines a set of supergroups.

The program Monte starts by reading in the highest energy cross-section band, and by arranging the memory for the highest energy output supergroup. The highest of the low-energy bounds of these energy ranges defines EBL, the low energy of the supergroup currently treated.

The program then calls the source-picking routine SOUPIC, and examines the energy E. If $E < EBL$, the particle is stored as a latent by calling the subroutine STORE, and SOUPIC is again called. When $E > EBL$, the subroutine CARLO is called. The subroutine CARLO tracks the particle, scores contributions to fluxes when needed, and distributes collisions along their track. Particles coming out of collision are also tracked if their energy is above EBL; particles coming out

of collision with E < EBL are stored as latents by calling STORE. Control is finally returned to MONTE, which proceeds to the next source particle, until a complete statistical aggregate of particles has been treated for the highest supergroup.

At this point, MONTE switches to the next supergroup by either reading a new band of cross-section data, or by writing out on tape the set of scores obtained and preparing the memory layout for the next supergroup, or both. It then proceeds to call a subroutine PICK, which picks latents from previous supergroups. If E < EBL the particle is stored again as a latent by calling STORE. If E > EBL, CARLO is called. The procedure continues until all latents have been examined, at which point MONTE switches to the next supergroup until the low-energy cutoff is encountered. When this occurs, MONTE switches to the highest supergroup, and proceeds to treat the next statistical aggregate of particles. The calculation terminates when a history number exceeds the cutoff value NHIST specified on input. A "blank" interaction record, with NHIST = NHIST + 1, is written on the interaction tape and all tapes are rewound. Control is transferred to TUNC which calls the editing program PEDIT.

8.2 SOUPIC – THE SOURCE-PICKING ROUTINE

This is a subroutine which picks particles from the biased source distribution.

If an external source tape is to be used, groups of 35 source particles are read from tape into a buffer, and returned one by one to the main code. The quantities describing a source particle are:

$\vec{XE}$ – coordinates of the particle
IR – region number where particle is born
$\vec{WB}$ – direction of the particle
T – time at which particle is born
E – energy of the particle
NHIST – history number attached to the source particle
F – statistical weight of the particle generated

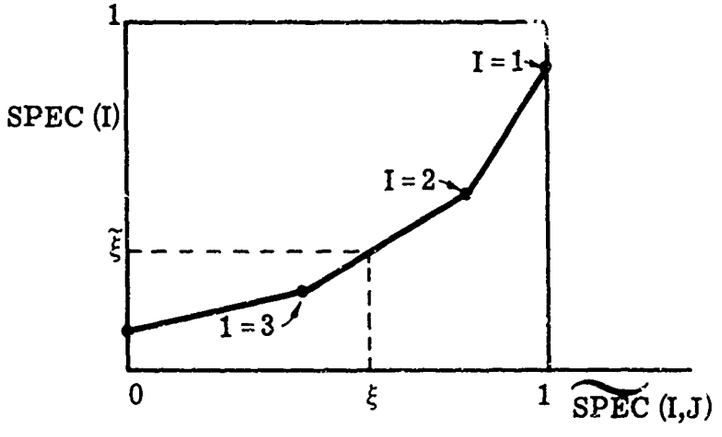
W_C – carry-along statistical weight of the particle (usually set equal to unity).

A first random number ξ is compared to the table SOUR(L). The smallest L for which SOUR(L) > ξ determines the region IR = ISR(L) to be picked from. Standard techniques are used to pick coordinates of points uniformly distributed in a region; coding is included for any region in either spherical or MASQUE geometry, for rectangular parallelepipeds (both ordinary and nonordinary), and spheres, without internal bodies, for the UNC-SAM geometry.

The energy is picked next. A stratified random number (called CE in the code) is obtained (stratification is done for each statistical aggregate of source particles). A biased random number $\tilde{\xi}$ is then obtained by interpolation in the SPEC vs $\widetilde{\text{SPEC}}$(J) tables pre-computed by SOUCAL. (The J is determined by the region number.) Finally the energy is determined by solving the equation

$$\tilde{\xi} = \int_E^\infty S(E)dE$$

using semi-log interpolation as explained in Chapter 7.



The direction $\overrightarrow{WB}$ of the source particle is determined as follows. If the source is isotropic and there is angular importance sampling, the cosine of the angle between the particular aiming angle and the direction is chosen by picking a random number, and interpolating between the angular mesh supplied on input vs the table P(I,K) pre-computed by SOUCAL. (The K is determined by the region number.) A random azimuth is then picked, which completes the specification of the direction. In the absence of angular importance in the source region, standard techniques are used. The case of a monodirectional source also can be handled, provided there is no angular importance in the source region. Finally, if time dependence is to be determined, a time T is determined by another random number ξ and the solution of the equation

$$\xi = \int_0^T S_t(t)dt.$$

The quantities communicated to the main code are $\overrightarrow{XB}$, IR, $\overrightarrow{WB}$, T, E, NHIST, F, and W_C (where $W_C = 1$), and F is the weight in region IR at energy E in the direction $\overrightarrow{WB}$.

8.3 CARLO – THE TRACKING AND SCORING ROUTINE

Source particles, either read from source tape or generated internally by the subroutine SOUPIC, as well as latents, are tested on energy. If the energy is below the lower bound of the supergroup currently treated, they are stored as latents. For energies within the supergroup, they are transmitted to the subroutine CARLO which tracks, distributes further collisions, if any, and scores answers.

Given the region number IR, the energy E and the direction of flight $\overrightarrow{WB}$, a total weight

$$W = W_X \times W_E \times W_{\vec{\Omega}}$$

is calculated and a new F given by the ratio F/W is treated as the total number of biased particles born simultaneously at $\overrightarrow{XB}$, T, with an energy E, in the direction $\overrightarrow{WB}$. The carry-along statistical weight, W_c, can be considered as a normalization factor, to be transmitted to all future collided particles due to the present source particle. For internally generated particles the quantity F will always be unity. Externally generated particles will have an F different from unity if they were not picked from the properly biased source distribution. The quantity F is tested vs an input cutoff value FZ($\leq$1). If it is smaller, a game of chance is played, and, with probability (1-F) the particle is killed; whereas with probability F the particle is kept with F set as unity. These two events are scored as either (F $\times$ W) "kills" or (1-F)$\times$W "births."

If the particle survives the test, a subroutine DR is called, which provides the total macroscopic cross section μ in the region IR (this routine is almost identical to the DR routine described in UNC-5093).[1] A random number ξ_1 is picked for picking future collision points.

A geometry routine, G1(S_1, IR′, $\vec{X}'$), is called to provide the distance S_1 to the first region boundary encountered from $\overrightarrow{XB}$ in the direction $\overrightarrow{WB}$. IR′ is the region number on the other side of the boundary, and $\vec{X}'$ is the point of intersection of the track with the boundary. If the region IR is a scoring region, the contribution to the flux is calculated and scored, as explained in Chapter 1. Points where particles enter into collision are picked by computing

$$\xi' = \xi_1 - F(1-e^{-\mu S_1}).$$

If $\xi' < 0$, a particle comes into collision in region IR at $\vec{X} = \overrightarrow{XB} + S'\vec{\Omega}$ where $S' = -1/\mu \log (1 - \xi_1/F)$; the interaction routine DR is called to pick a particle coming out of collision. The interaction is written on tape, and the particle coming out of collision (if ever) is stored as a latent. ξ_1 is then increased by one, and control is returned to the part of the code which computes ξ' to attempt to produce

more collisions in region IR. When $\xi' > 0$, no more collisions are produced in region IR. F is then set to $F \times e^{-\mu S_1} \times W$, where W is the weight. A test is made whether the next region is a transmission region (if it is, the coordinates $\vec{X}'$, IR, energy, time, etc. are written on tape) and whether it is the escape region. If it is not an escape region, the particle tracked is moved to the boundary by setting $\overrightarrow{XB} = \overrightarrow{XP}$, IR = IR', etc., by computing the new weight W, and by setting a new F equal to F/W. If $F < FZ$, the particle is either killed or F is set equal to 1, both events being properly tallied. For $F < FZ$, ξ is set equal to ξ', the DR routine is called to obtain the new total cross section μ, and control is transferred to the part of the code which calls the geometry routine G1 (S_1, IR', $\vec{X}'$), thus the tracking continues until either a kill occurs or the escape region is reached. In the latter case F "escapes" are tallied.

We mentioned above that particles coming out of collision (if ever) were stored as latents. An absorbed particle does not come out of collision, but is scored as W "absorbs." Elastic or inelastic particles come out with some energy E. If E is less than the low-energy bound of the supergroup currently treated, the particle is stored until the next supergroups are treated by calling the subroutine STORE. If the energy is high enough, the particle is stored in a local buffer.

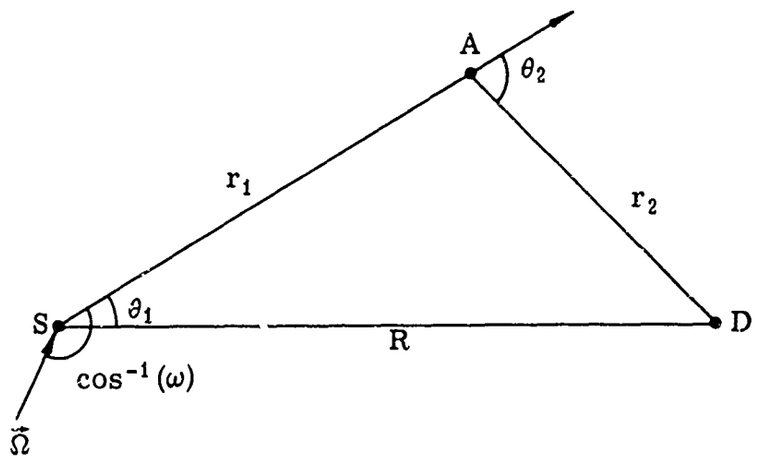
Upon termination of a tracking, the subroutine CARLO examines this local buffer, and, if any latents are available, picks one and performs the tracking. When the buffer is empty, control is returned to the MONTE program.

8.4 FAP – THE FLUX-AT-A-POINT ROUTINE

The question of estimation of flux-at-a-point by Monte Carlo methods was investigated in detail by M. H. Kalos.[5]

The methods he proposed deal with obtaining a distribution of points where particles enter into collision with normal Monte Carlo methods, and estimating the

once-more-collided flux, at the point of interest, using a special distribution of the last collision point.



Assuming a collision point at S, a detector at D as shown in the sketch, the expression for the once-more-collided flux is

$$\varphi = \int g_S(\omega) \frac{e^{-\int_0^{r_1} \mu ds}}{4\pi r_1^2} \mu_A g_A(\cos \vartheta_2) \frac{e^{-\int_0^{r_2} \mu ds}}{4\pi r_2^2} dV \qquad (13)$$

where $g_S(\omega)\,d\omega$ is the probability of scattering through an angle whose cosine is in the range $(\omega, \omega + d\omega)$.

For isotropic source particles one has to estimate in addition the uncollided flux.

One of the methods Kalos proposes is to pick the intermediate points A from a distribution given by

$$\frac{R}{\pi^3} \frac{dV}{r_1^2 r_2^2} \qquad (14)$$

the estimator becoming

$$f_1 = \frac{\mu_S \pi}{16R}\, g_S(\omega)\; g_A(\cos\theta_2)\, e^{-\int_0^{r_1} \mu ds}\; e^{-\int_0^{r_2} \mu ds} \tag{15}$$

This is the method used in the present code. It turns out that there is a very simple way to pick intermediate points A from the distribution (Eq. 14), if a proper choice of coordinate system is made.

Let us define the point A by the three angles θ_1, θ_2, and φ, where θ_1 and θ_2 are defined on the sketch and φ is an azimuth around SD.

The distribution (Eq. 14) becomes:

$$\frac{R}{\pi^3} \frac{r_1^2 \sin\theta_1\, d\theta_1\, d\varphi \frac{\partial r_1}{\partial \theta_2}\, d\theta_2}{r_1^2 r_2^2} \tag{16}$$

Using the relationships:

$$\frac{r_1}{\sin(\theta_2 - \theta_1)} = \frac{r_2}{\sin\theta_1} = \frac{R}{\sin\theta_2}$$

one has

$$r_2 = \frac{\sin\theta_1}{\sin\theta_2} R \quad \text{and} \quad \frac{\partial r_1}{\partial \theta_2} = \frac{\sin\theta_1}{\sin^2\theta_2} R.$$

Substituting into Eq. 16 one obtains the distribution

$$\frac{1}{\pi^3}\, d\theta_1\, d\theta_2\, d\varphi \tag{17}$$

i.e., a random distribution of θ_1, θ_2, φ in the range

$$0 \leq \theta_1 \leq \theta_2 \leq \pi \tag{18}$$

$$0 \leq \varphi < 2\pi \tag{19}$$

Once θ_1 and φ are picked, the cosine of the angle of scattering ω is easily obtainable, and the score (Eq. 15) can be evaluated.

In the actual program, coding is provided for either the procedure outlined above, which gives a finite variance provided the detector is not in the immediate vicinity of the source region, or for a simple statistical estimation method, which gives finite variance if the detector is not in the immediate vicinity of either the source region or of a scattering material. In that case the quantity

$$g_S(\cos\theta)\frac{e^{-\int_0^R \mu ds}}{4\pi R^2}$$

is scored for each collision point S, where θ is the angle between the directions $\vec{\Omega}$ and $\overrightarrow{SD}$. The type of calculation to be performed is specified on input, by specifying the total number of detectors ND, and the number NDFAP of detectors for which the first procedure is to be applied. The remaining (ND-NDFAP) detectors will be treated by statistical estimation.

If ND $\neq$ 0, the flux-at-a-point routine FAP is called for each source particle by the MONTE program, and for each collision point by the CARLO routine, except if inelastic scattering occurs giving an outgoing energy below the supergroup being currently treated. In that case the particle is stored as a latent, and MONTE will call the FAP routine with that latent when the proper supergroup will be treated.

Once called, the subroutine FAP examines whether the particle to be treated is a source particle, a particle coming out of inelastic scattering, a particle coming out of elastic scattering, or a latent from a previous FAP calculation. In the first

three cases, the trigonometric functions of θ_1 and θ_2 picked according to the distribution (Eq. 17) are obtained. This is done rather efficiently by picking directly the sin and cos functions of two random angles, and then determining which corresponds to the smallest angle, and therefore which should be assigned to θ_1; the others are assigned to θ_2. θ_1 and θ_2 are kept the same for all FAP-detectors in the problem.

For source particles, the uncollided flux is computed (the source is assumed to be isotropic).

The contribution of the once-more-collided flux is computed as follows. The distances r_1 and r_2 are given by $r_2 = R \sin \theta_1 / \sin \theta_2$ and $r_1 = R \sin \theta_1 - r_2 \sin \theta_2$. A random azimuth around $\overrightarrow{SD}$ is picked. For both source particles and inelastic particles, $g_S = 1$, and the outgoing energy is as set by the main code (either energy of the source particle, or energy of the inelastic particle). (No energy-angular correlation is assumed in the code. The difference between lab and center-of-mass system is also ignored for inelastic scattering.) For particles coming out of elastic scattering, the angle of scattering is computed in the lab system. A subroutine GE is called. For neutrons, it performs the transformation to the CM system, looks up the probability g_S, and computes the outgoing energy E. For gamma rays, g_S and E are given by the Klein-Nishina formula. If this energy is below the supergroup currently treated, the particle is stored as a latent of the first kind to be picked up and returned to this point of the coding at a later time in the calculation.

The subroutine DR is called, and the total cross section μ is made available. A subroutine TRALA is called, which tracks the particle a distance r_1 in the direction of flight, with an energy E, and computes

$$\lambda_1 = \int_0^{r_1} \mu ds;$$

if the particle escapes, an "infinite" λ_1 ($\lambda_1 = 27$) is returned, no score is made, and the next (if any) detector is treated. If the particle did not escape, the subroutine DR is called to determine the type of interaction occurring at the intermediate point. Absorption corresponds to no score. For inelastic particles, E is picked from the spectrum by the DR routine, and $g_A = 1$. For elastic scattering, the subroutine GE determines g_A and E. If the energy E is below the supergroup currently treated, the particle is stored as a latent of the second kind, to be picked up and returned to this point of the coding at a later time in the calculation. Finally, the subroutine TRALA provides

$$\lambda_2 = \int_0^{r_2} \mu ds,$$

and the total contribution is calculated and scored, and the next detector (if any) is considered, unless the calculation dealt with a latent of either first or second kind, which applies only to a particular detector. When all FAP detectors have been considered, the statistical estimation detectors are treated by using the coding described above for the tracking from intermediate point to the detector, where the intermediate point is replaced by the original collision point.

8.5 PICK AND STORE – THE TREATMENT OF LATENTS

We have seen throughout the previous sections that particles degrading below the energy EBL, low-energy limit of the supergroup currently treated, were stored as latents by calling the subroutine STORE. They were later picked up by MONTE by calling the subroutine PICK. The two subroutines are actually a single one with two entry points. The subroutine INPUTD allocates the memory to data, scores, etc. The remaining memory is assigned to the subroutine PICK-STORE, to be used as a buffer for latents. One "end" of the buffer is assigned to "degraded" particles, this is the end of the buffer where particles are being stored. The other "end" of the buffer is assigned to "unsorted" particles, i.e.,

the particles to be picked. Associated to each end of the buffer, there is a magnetic tape to be used when the buffer overflows. There are two modes of operation. In one mode, the top of the buffer is unsorted and the bottom is sorted. When the switch is made from one supergroup to the next, the "unsorted" part is empty, and the "degraded" part may have particles which become "unsorted" for the supergroup about to be treated. The designation of the buffers (and of the tapes) is therefore switched.

There is no set boundary between the two "ends" of the buffers. The number of particles in the "unsorted" buffer keeps decreasing, whereas the number of particles in the "degraded" buffer keeps increasing, and can increase faster than the other number decreases. Therefore, the two parts of the buffer can meet, causing an overflow of the buffer.

It is then determined which "end" of the buffer is longest, and a number of particles exactly equal to one-half the total length of the buffer are written from the longest "end" onto the corresponding magnetic tape.

When the "unsorted" buffer becomes empty, a test is made whether any "unsorted" particles are available on the corresponding magnetic tape. If none are available, the calculation has been completed for the current supergroup. If some are available, they are read into the buffer if room is available. If room is not available, it is made available by writing out part of the other buffer on the other tape; the length of the record written out from one "end" is equal to the length of the record to be read into the other "end."

The subroutine PICK-STORE deals with different kinds of latents. The quantities stored are: $\vec{X}$, $\vec{\Omega}$, E, IR, T, I, F, NHIST, WC, and J12345, where $\vec{X}$ is the position, $\vec{\Omega}$ the direction, E the energy, IR the region number, T the time, F the weight, NHIST the history number, and WC a normalization factor. I and J12345 are indices.

J12345 = 1 - identifies a source particle

= 2 - identifies a particle coming out of elastic scattering

= 3 - identifies a particle coming out of inelastic scattering

= 4 - identifies a latent of the second kind for FAP only

= 5 - identifies a latent of the first kind for FAP only.

(In other parts of the code J12345 = 6 identifies a transmitted particle, J12345 = 7 an inelastic interaction, and J12345 = 8 an absorption.)

I is irrelevant (set to 0) for J12345 = 1, 2, 3 (and 6). For FAP latents (J12345 = 4, 5), I is set to IDET, the detector number for which the latent applies. [In the description of interactions (J12345 = 7, 8), I is set to IATWT, a five-digit identifier of the element with which the interaction occurred.]

When the program MONTE calls the subroutine PICK, it examines the J12345 obtained and calls the proper routines:

J12345 = 1,3 - both FAP and CARLO

= 2 - CARLO only

= 4,5 - FAP only.

9. PEDIT – THE EDIT PROGRAM

The PEDIT program runs through the output tape, calculates average fluxes and standard deviations, and prints them out in a self-explanatory format. It also gives a tally of the absorptions, kills, births, and escapes occurring in different geometrical regions.

10. RESPONSE

The program RESPONSE is a separate program which accepts as input the output tape of an UNC-SAM-2 problem and card input to provide response functions $R_{ijk}(E)$. It computes

$$R_{ijk} = \int_0^E \varphi_{ik}(E) R_{ij}(E) dE$$

in specified regions i, for all time bins k, for all response functions j. It also computes associated statistical errors. $\varphi_{ik}(E)$ is assumed to be equal to the average flux $\varphi_{ik\ell}$ to energies E in the ℓ^{th} bin.

An input sheet is shown in Chapter 13.

A first card gives:

NHIST	Number of histories to be treated (must be ≤ the number of histories treated by the Monte Carlo code).

The next card gives:

NRESP	Number of response functions in the problem

The description of each response function is as follows. A first card gives:

NEN	Number of entries in the table
NREG	Number of regions in which response is to be calculated
50H	50 Hollerith characters for identification purposes.

The next cards specify the response function R(E). Pairs of numbers E, R(E) are inputed. Linear interpolation is assumed between entries in the table.

Finally, a list of the (NREG) region numbers in which the response is to be calculated should be entered.

11. GASP – THE SECONDARY PARTICLE GENERATING PROGRAM

This program incorporates simple interaction routines other than those built into the code. Given as input points where particles go into interaction written on a magnetic tape during a previous UNC-SAM-2 calculation, given a simplified description of the interaction, and a description of the statistical weights, GASP generates biased particles coming out of collision and writes them on another magnetic tape to be used as a source tape for a future UNC-SAM-2 calculation. The main use of GASP is for generating inelastic and capture gamma rays due to neutron interaction in matter. It can also be used for generating the second neutron from (n,2n) reactions, fission neutrons, etc.

An input form is shown in Chapter 13.

Input

A first card specifies:

NELE	Number of elements present in the configuration, and for which GASP data are provided
IRMAX	Number of geometrical regions in the problem
IHCUT	Maximum number of histories to be treated
IGCUT	Maximum number of secondary particles to be created
NOUT	Number of energy bins for the edit of the interaction tape.

If NOUT > 0, this is followed by a specification of the (NOUT + 1) energy bounda-

ries of the NOUT energy bins for an edit of the interaction tape. If NOUT = 0, no bin input is required and the edit is bypassed.

If IGOUT > 0, this is followed by the GASP element data for the NELE elements.

For each element, this consists of:

1. Element Parameter Card

IATWT	Integral part of the atomic mass (identification of element same as that specified in DATORG input).
L_A	Number of absorption gamma output energies (0 if no capture gamma produced).
K_A	Number of corresponding incident neutron energies. This equals the number of neutron energy bins (0 for no gamma capture).
L_I	Number of inelastic gamma out energies.
K_I	Number of corresponding incident neutron energies. This equals the number of neutron energy bins + 1.

2. Capture Gamma Energy Cards (as many as necessary)

The energies are given in ascending order in eV.

3. Capture Neutron Energy Cards (as many as necessary)

The entries are in ascending order and in eV and represent the upper energy of an energy bin (the low energy of the lowest energy bin is not entered, and is assumed to be zero).

4. Capture Number Cards

Each entry represents the number of gammas produced at each of the specified gamma energies. These numbers are entered for each neutron energy bin.

Start a new card for each neutron bin.

Repeat items 2, 3, and 4 for inelastic scattering input, if any.

The element data input is followed by a description of the region and energy weights for the secondary particles, in a format identical to that of the UNC-SAM-2 code. Note that angular-dependent weights are not entered.

If IGCUT = 0, no element data, and no weight specifications are required. The generation of secondaries is bypassed.

The following cards specify the energy bin structure for the edit of a source tape (the source tape currently generated if IGCUT > 0, or an existing source tape of IGCUT = 0).

A card gives:

NOUT Number of output bins.

This is followed by cards specifying the (NOUT+1) energy bins boundaries, in decreasing order.

This is the end of card input. The code also requires as input an interaction tape, and a source tape (if IGCUT≠0).

Output

The output consists of the following.

1. If NOUT ≠ 0 on the first card, an edit of the interaction tape is printed. This lists the number of interactions occurring for each geometrical region for each element for each energy bin. If NOUT = 0, this edit is bypassed.
2. If IGCUT ≠ 0, the element data and the weight specification are printed back.

The code then proceeds to generate secondary particles and writes them on the source tape. This is done by reading a record of a particle which entered into an interaction. The record consists of $\vec{X}$, $\vec{\Omega}$, E, T, NHIST, F, W_C, IR, J12345, IATWT (where J12345 is an integer which specifies whether the interaction was an inelastic scattering or an absorption), and IATWT is a five-digit integer identifier of the element with which the interaction occurred. The code proceeds to look up the data for element IATWT for the energies E_γ and the number $n(E_\gamma)$ for each energy, of secondary particles to be created. For each energy, it computes a weight

$$W = W_X(IR)\, W_E(E)$$

and computes the effective number

$$\tilde{n}(E_\gamma) = n(E_\gamma)\frac{F}{W}$$

of secondaries to be picked from the biased kernel. The fractional part of this number is then made either zero or unity by generating a random number ξ and truncating

$$\tilde{N} = \tilde{n} + \xi.$$

$\tilde{N}$ directions $\vec{\Omega}'$ are picked isotropically, and $\tilde{N}$ source particles are written on the source tape, all at the same $\vec{X}$, the same E, the same T, the same NHIST, the same F set equal to the weight W, the same W_C, the same IR, but with the $\tilde{N}$ different directions $\vec{\Omega}'$.

When all E_γ's have been treated for the given interaction, the code proceeds to the next, unless NHIST > IHCUT, or the total number of secondaries generated exceeds IGCUT, or the last ("blank") interaction has been reached. At this point the code writes a "blank" source particle with NHIST = NHIST + 1.

If IGCUT = 0, all of this calculation is bypassed, but a previously generated source tape is assumed to be mounted.

The code then rewinds the source tape and edits it, giving the number of source particles for each region for each energy bin for all source particles up to, but excluding, the last "blank" one.

As can be seen, the program GASP ignores completely the existence of possible angular-dependent weights. If no such biasing is used, GASP picks from the exact modified collision kernel. If angular importance is used in the future UNCSAM calculation for the transport of the secondaries, the source particles are not generated from the completely modified kernel. However, upon being picked, the weight $F = W_X W_E$ of the source particle generated is divided by the total weight $W = W_X W_E W_{\vec{\Omega}}$ of the particle, giving an F different from unity, and the angular importance of the particle picked will be correctly treated in the transport problem. In other words, this means that the sampling is not as good as it could be, but that the answers are still correct, on the average.

The running of the subsequent SAM-2 problem is straightforward. The region and energy weights should be kept, if possible, the same as those specified in GASP, although the argument similar to the one above shows that there is no great harm in their modification.

The statistical aggregate size for the secondary problem should be the same as, or a multiple of, that of the primary problem. This is due to two reasons. If the primary problem was run using the internal source generator, the energy of the source particles was picked in a stratified and ordered (decreasing E) way. Furthermore, if the primary problem was run with more than one energy supergroup, the collisions corresponding to a given statistical aggregate are not necessarily ordered in history number.

12. TRANSMIT – TREATMENT OF TRANSMITTED PARTICLES

As mentioned earlier in the description of CARLO, one can define a "transmission" region. Particles crossing into that region are written on the interaction tape with a special index (J12345=6) identifying them.

The program TRANSMIT is a rather simple one. It runs through the interaction tape, and, each time it encounters a transmitted particle, writes it out on a source tape. It also edits this tape provided that output bins are specified. (See input sheet in Chapter 13.)

13. INPUT FORMS

TUNC INPUT

78 COLUMNS OF HOLLERITH IDENTIFICATION	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	79 80

IBAND	NBAND
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20

BAND INPUT
(Fill out only if IBAND = 0)

PROBLEM IDENTIFICATION CARD (TO BE FILLED OUT ONCE ONLY)			No. of distinct compositions	NG NG = 0 neutrons NG = 1 γ-rays	NBAND		Deck I.D. No.*
Problem number	Upper energy of cross-section data, eV	Lower energy of cross-section data, eV					
Columns 1–10	11–25	26–40	41–45	46–50	51–55	56–72	73–80
	• E+	• E+					XSDAT 02

Note: Energy limits for identification purposes only.

Enter NBAND + 1 numbers

CROSS-SECTION BAND LIMITS (HIGH TO LOW)						
Columns 1–14	15–28	29–42	43–56	57–70	71–72	73–80
• E+	• E+	• E+	• E+	• E+		XSDAT 03

(Use as many cards as necessary)

*Every card is to be numbered sequentially.

BAND INPUT – (CONTINUED)

REPEAT FOLLOWING FOR EACH COMPOSITION

COMPOSITION I.D. CARD

Composition number	Number of elements		Deck I.D. No.*
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17	18 19 20	73 74 75 76 77 78 79 80
			X S D A T

ELEMENT CARDS

Atomic weight integral point only	Concentration = $\frac{\rho N_0}{A}$ × volume fraction × 10^{24}		Deck I.D. No.*
1 2 3 4 5	6 7 8 9 10 11 12 13 14	15 16 17 18 19 20	73 74 75 76 77 78 79 80
	•	E	X S D A T
	•	E	X S D A T
	•	E	X S D A T
	•	E	X S D A T
	•	E	X S D A T

*Every card is to be numbered sequentially.

GEOM INPUT

TITLE CARD

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

ITYPE	NRGN	NSRF	IREX	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

GEOM INPUT – (CONTINUED)
ITYPE = 1 – MASQUE Geometry

1. Line input. Enter information for I lines (I = 1, NSRF)

R_1	Z_1	R_2	Z_2	I	
				1	
				2	
				3	

2. Region description. Enter information for region IR (IR = 1, NRGN)

L_1	L_2	L_3	L_4	J_1	J_2	J_3	J_4		IR	

For each negative J enter (-J) pairs of numbers, using as many cards as necessary.

IR'_1	Q_1	IR'_2	Q_2	IR'_3	Q_3	IR'_4	Q_4
IR'_5	Q_5		etc.				

GEOM INPUT – (CONTINUED)
ITYPE = 2 – Spherical Geometry

Enter the NRGN radii in increasing order. Use as many cards as necessary.

1 2 3 4 5	6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
R_1		R_2	R_3	R_4	R_5	R_6	
R_7		R_8	etc.				

GEOM INPUT – (CONTINUED)

ITYPE = 3 – UNC-SAM Geometry

PROBLEM INFORMATION CARD

Number of ordinary rectangular parallelepipeds, M	Number of non-ordinary regions, N		Card I.D. No.
1 – 6	7 – 10	11 – 20	21 – 72	73 – 80
				E Z G E O M 2

Note: M + N ≤ 200
N ≤ 127

BOUNDING PLANE CARDS FOR ORDINARY RECTANGULAR PARALLELEPIPEDS (ONE CARD PER REGION)

X_α coordinate for region	X_β coordinate for region	Y_α coordinate for region	Y_β coordinate for region	Z_α coordinate for region	Z_β coordinate for region	Card I.D. No.* (No. sequentially)
1 2 3 4 5 6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O
• E	• E	• E	• E	• E	• E	B P O

*This number should be the number of the pertinent region.

SPHERICAL GEOMETRY INPUT (ONE CARD PER SPHERE)

X coordinate of center of sphere (cols 1–12)	Y coordinate of center of sphere (cols 13–24)	Z coordinate of center of sphere (cols 25–36)	Radius of sphere (cols 37–48)	(cols 49–52)	(cols 53–55)	(cols 56–57)	Number of the region that is external to sphere (cols 58–60)	(cols 61–72)	(cols 73–75)	Region number of sphere (cols 76–80)
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	
. E	. E	. E	. E		SPH				SPH	

CYLINDRICAL GEOMETRY INPUT (CARD 1)

X coordinate of center of base of cylinder	Y coordinate of center of base of cylinder	Z coordinate of center of base of cylinder	Radius of base of cylinder		Number of the region that is external to cylinder		Region number of cylinder
1 2 3 4 5 6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55	56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
●E	●E	●E	●E	CYL			CYL

CYLINDRICAL GEOMETRY INPUT (CARD 2)

X component of the height vector	Y component of the height vector	Z component of the height vector		Region number of cylinder
1 2 3 4 5 6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
●E	●E	●E		CYL

Repeat above 2 cards for each cylinder

WEDGE GEOMETRY INPUT (CARD 1)

X coordinate of vertex of wedge	Y coordinate of vertex of wedge	Z coordinate of vertex of wedge			Number of region that is external to wedge		Region number of wedge
1–12	13–24	25–36	37–52	53–55	56–57	58–60	61–72	73–80
F	F	F		W,E,D			W,E,D

WEDGE GEOMETRY INPUT (CARD 2) *

X component of first base vector	Y component of first base vector	Z component of first base vector	X component of second base vector	Y component of second base vector	Z component of second base vector	Region number of wedge
1–12	13–24	25–36	37–48	49–60	61–72	73–80
F	F	F	F	F	F	W,E,D

WEDGE GEOMETRY INPUT (CARD 3)

X component of the length vector of wedge	Y component of the length vector of wedge	Z component of the length vector of wedge		Region number of wedge
1–12	13–24	25–36	37–72	73–80
F	F	F		W,E,D

*See Section 6.1

Repeat above 3 cards for each wedge

NON-ORDINARY RECTANGULAR PARALLELEPIPED GEOMETRY INPUT (CARD 1)

X coordinate of vertex	Y coordinate of vertex	Z coordinate of vertex			Number of region that is external to the region			Region Number
1 2 3 4 5	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52	53 54 55	56 57	58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
●E	●E	●E			RPP				RPP

NON-ORDINARY RECTANGULAR PARALLELEPIPED GEOMETRY INPUT (CARD 2)

X component of first base vector	Y component of first base vector	Z component of first base vector	X component of second base vector	Y component of second base vector	Z component of second base vector	Region number
1 2 3 4 5	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
●E	●E	●E	●E	●E	●E		RPP

NON-ORDINARY RECTANGULAR PARALLELEPIPED GEOMETRY INPUT (CARD 3)

X component of the length vector	Y component of the length vector	Z component of the length vector		Region number
1 2 3 4 5	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
●E	●E	●E		RPP

Repeat above 3 cards for each non-ordinary rectangular parallelepiped

FRUSTA OF CONES GEOMETRY INPUT (CARD 1)

X coordinate of lower base	Y coordinate of lower base	Z coordinate of lower base	Radius of lower base			Number of the region that is external to frustum			Region number of frustum
1 2 3 4 5	6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52	53 54 55	56 57	58 59 60	61 62 63 64 65 66 67 68 69 70 71 72	73 74 75	76 77 78 79 80
•	E	•	E	•	E			C N E		C N E	

FRUSTA OF CONES GEOMETRY INPUT (CARD 2)

hX component of the height vector	hY component of the height vector	hZ component of the height vector	Radius of upper base			Region number of frustum
1 2 3 4 5	6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80
•	E	•	E	•	E	C N E

Repeat above 2 cards for each frustum

INPUTD INPUT

General Information

NSTART	NSTOP	NSTAT	NRMAX	NG	NT	NOUT	NUMSC	NRWL	IREX	IRT	
1-10	11-20	21-30	31-35	36-40	41-45	46-50	51-55	56-60	61-65	66-70	71-80

Point Detector Input

NDET	NDFAP	
1-10	11-20	21-80

Enter coordinates of the NDET detectors, one card per detector – flux-at-a-point detectors first, then statistical estimation detectors

XAD	YAD	ZAD	
1-16	17-28	29-42	43-80
E+	E+	E+	
E+	E+	E+	
E+	E+	E+	

INPUTD – (CONTINUED)

Cutoff Information

ECUT	ETH	TCUT	FZ	EHIGH	
1 2 3 4 5 6 7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48 49 50 51 52 53 54 55 56	57 58 59 60 61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
E+	E+	E+	E+	E+	

Output Energy Bin Specification. NOUT + 1 energy bin boundaries, from high to low – the first, last (and intermediate boundaries when supergroups are used) must be negative.

1 2 3 4 5 6 7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48 49 50 51 52 53 54 55 56	57 58 59 60 61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	

INPUTD – (CONTINUED)

Output Time Bin Information (Omit if NT = 0. If NT > 0 enter NT + 1 time bin limits, from high to low. Last number must be 0.)

1 2 3 4 5	6 7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48 49 50 51 52 53 54 55 56	57 58 59 60 61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
E+	E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	E+	

Region Weights – enter NRWL numbers

1 2 3 4 5	6 7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48 49 50 51 52 53 54 55 56	57 58 59 60 61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
E+	E+	E+	E+	E+	E+	
E+	E+	E+	etc.			

Region Specification – use NRMAX cards

ISC	NREG	IRW	IEW	IAN	IANG	
1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
						for IR = 1
						for IR = 2
						etc.

INPUTD – (CONTINUED)

Energy Weight Specification

NEWL	NEW	
1 – 10	11 – 20	21 – 80

Energy Mesh (Omit if NEWL = 0. If NEWL > 0 enter NEWL + 1 energy bin boundaries in decreasing order)

1 – 14	15 – 28	29 – 42	43 – 56	57 – 70	71 – 80
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	

Enter "NEW" Energy Weight Sets – Each set consisting of NEWL weights. Begin a new card for each set.

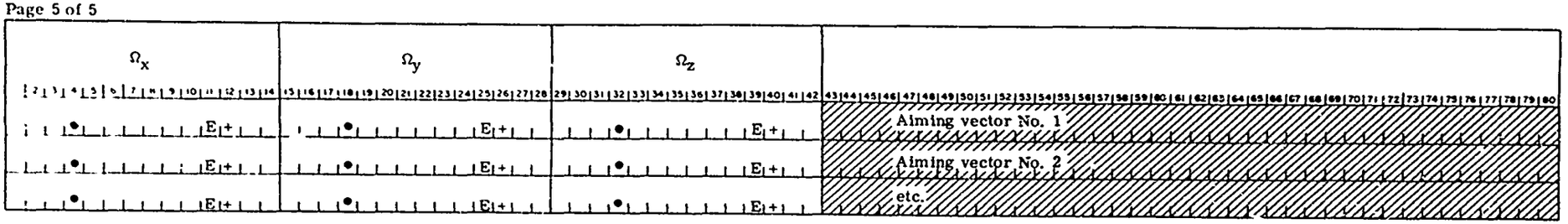
1 – 14	15 – 28	29 – 42	43 – 56	57 – 70	71 – 80
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	

Angular Weight Specification

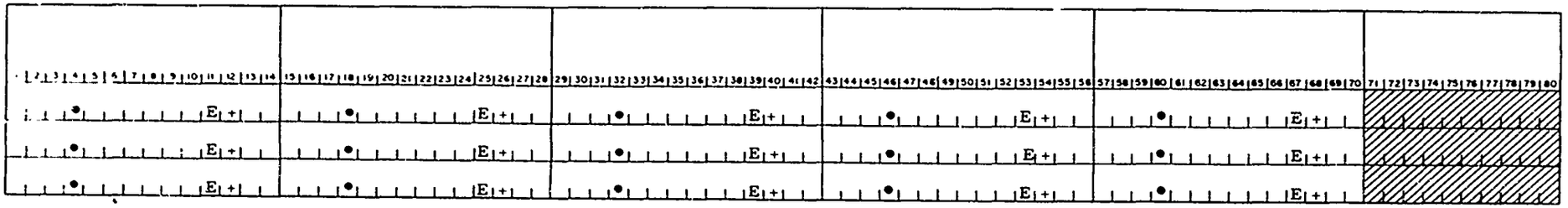
NAIML	NUMANL	NUMANG	
1 – 10	11 – 20	21 – 30	31 – 80

INPUTD – (CONTINUED)

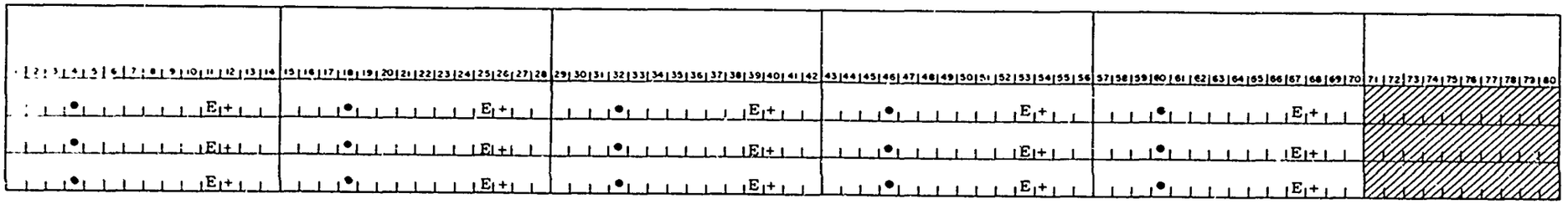
Aiming Angles – enter NAIML cards



Angular Mesh. Omit if NUMANL = 0. If NUMANL > 0 enter NUMANL + 1 angular bin boundaries in decreasing order. First number must be 1., last number must be –1.



Enter NUMANG Angular Weight Sets – each set consisting of NUMANL numbers. Begin a new card for each set.



SOURCE INPUT

NSR	IFLAG	ISW	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

End of Source Input if NSR = 0.
Source Regions Description – enter NSR cards.

ISR	P – Power Density	ISO	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	E+		
	E+		
	E+		

Spectrum Description – enter IFLAG cards

E (high to low)	$\int_E^\infty S(E)\,dE$	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
E+	E+	
E+	E+	
E+	E+	

Time Dependence Description – Omit if NT = 0

NOT	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

SOURCE – (CONTINUED)

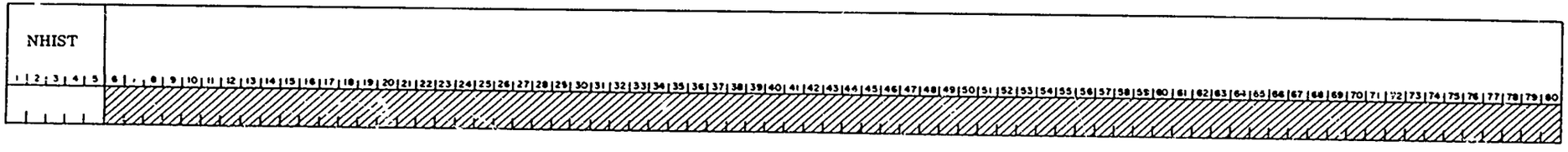
Enter "NOT" Cards

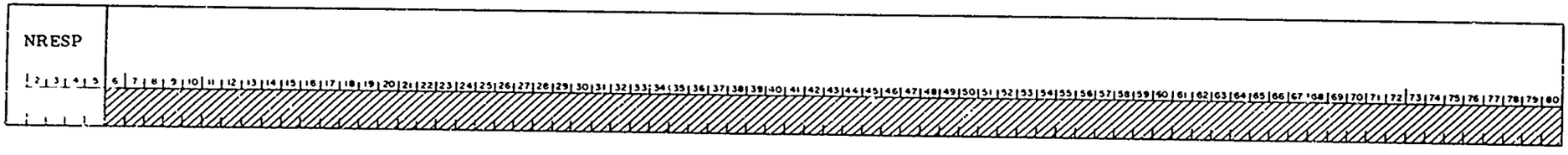
T	$\int_0^T S_t(t)\,dt$	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
E+	E+	
E+	E+	
E+	E+	

If ISO ≠ 0, enter direction cosines of source particles to be used in all regions

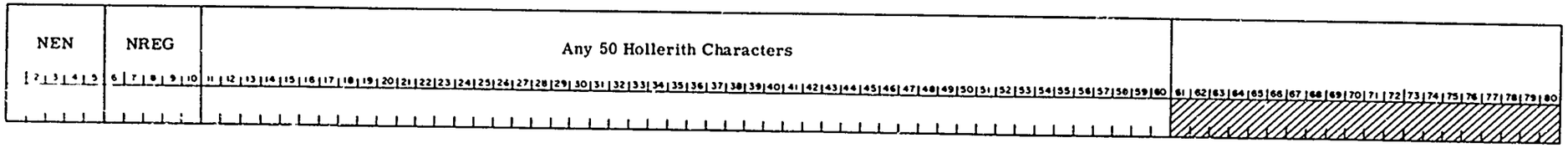
Ω_x	Ω_y	Ω_z	
1 2 3 4 5 6 7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
E+	E+	E+	

RESPONSE INPUT





Repeat the following for each of the NRESP response functions



RESPONSE – (CONTINUED)

Give NEN pairs (energy, response) in order of decreasing energy

ENERGY	RESPONSE	ENERGY	RESPONSE	ENERGY	RESPONSE	
1 2 3 4 5 6 7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36	37 38 39 40 41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
•E+	•E+	•E+	•E+	•E+	•E+	
•E+	•E+	•E+	•E+	•E+	•E+	
•E+	•E+	•E+	•E+	•E+	•E+	

Give the NREG region numbers, IR, in which the response is to be calculated

1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 50	51 52 53 54 55	56 57 58 59 60	61 62 63 64 65	66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

GASP INPUT
General Specifications

NELE	IRMAX	IHCUT	IGCUT	NOUT	
1 | 2 | 3 | 4 | 5	6 | 7 | 8 | 9 | 10	11 | 12 | 13 | 14 | 15	16 | 17 | 18 | 19 | 20	21 | 22 | 23 | 24 | 25	26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80

Output Bins for Interaction Tape Edit. Omit if NOUT = 0. If NOUT > 0, enter (NOUT + 1) energy bin boundaries, in decreasing order.

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14	15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28	29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42	43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56	57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70	71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	

Element Data Input (omit if IGCUT = 0). For each of the NELE elements, enter the following:

1. Element Parameter Card

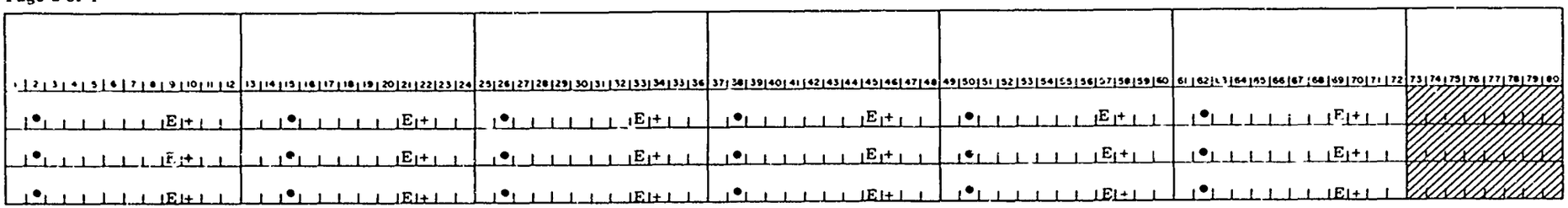
IATWT	LA	KA	LI	KI	
1 | 2 | 3 | 4 | 5	6 | 7 | 8 | 9 | 10	11 | 12 | 13 | 14 | 15	16 | 17 | 18 | 19 | 20	21 | 22 | 23 | 24 | 25	26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80

3. Capture Gamma Energy Cards.
   Enter L_A energies in increasing order

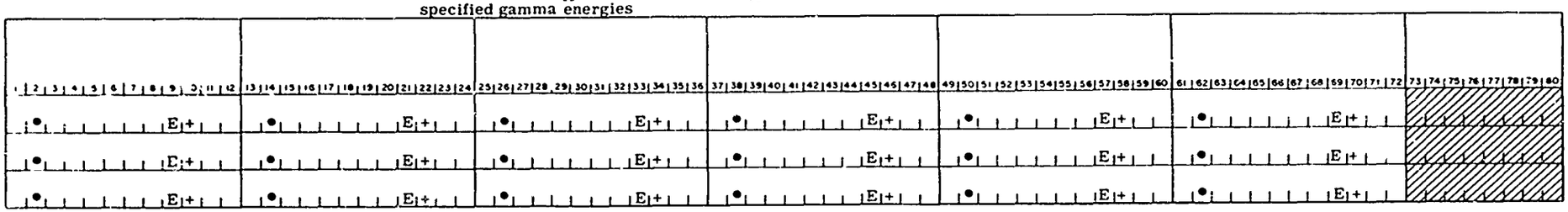
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12	13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24	25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36	37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48	49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60	61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72	73 | 74 | 75 | 76 | 77 | 78 | 79 | 80
E+	E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	E+	
E+	E+	E+	E+	E+	E+	

3. Capture Neutron Energy Cards
   Enter K_A upper bounds of neutron energy bins, in ascending order
   (Lower bound of low energy bin is implied to be 0.)

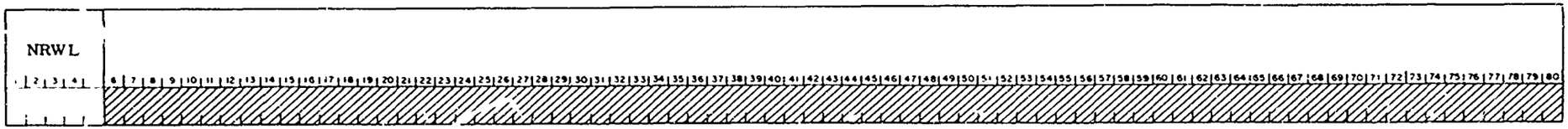
Page 2 of 4



4. Capture Number Cards
   For each of the K_A energy bins, enter L_A numbers of gammas produced at each of the specified gamma energies

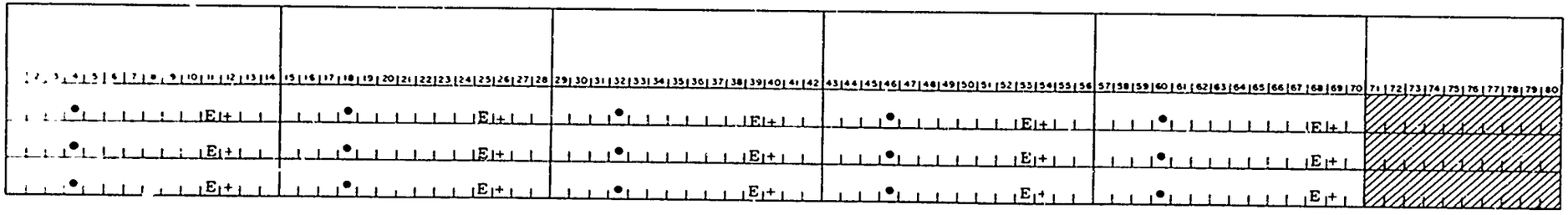


5. Repeat Items 2, 3, and 4 for inelastic scattering output, if any.
   After completion of element data – enter description of statistical weights

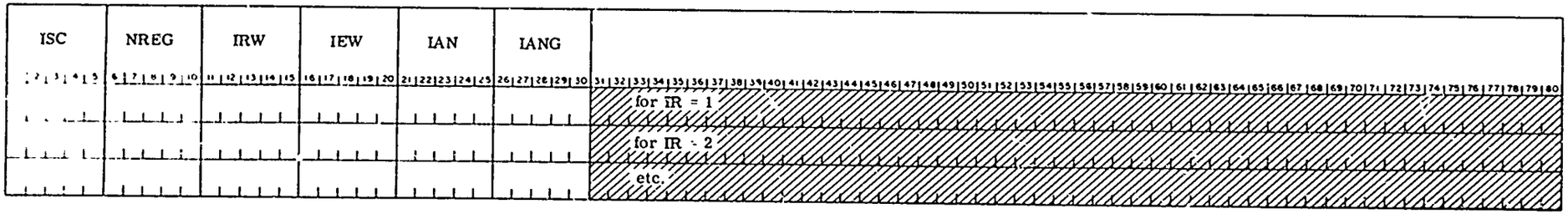


GASP – (CONTINUED)

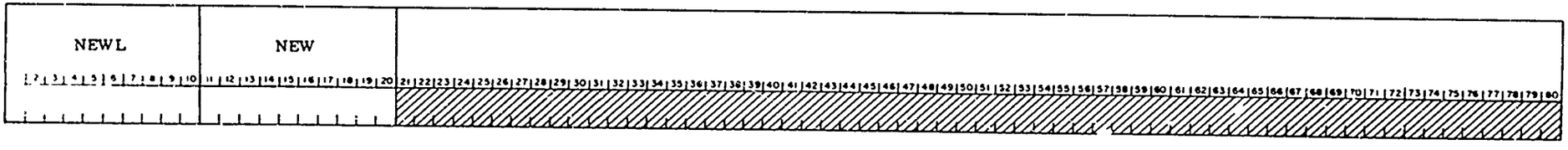
Region Weights – enter NRWL numbers



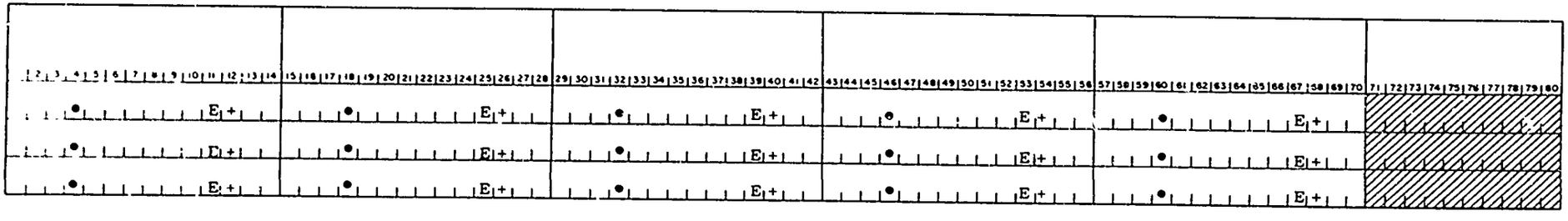
Region Specification – use NRMAX cards



Energy Weight Specification



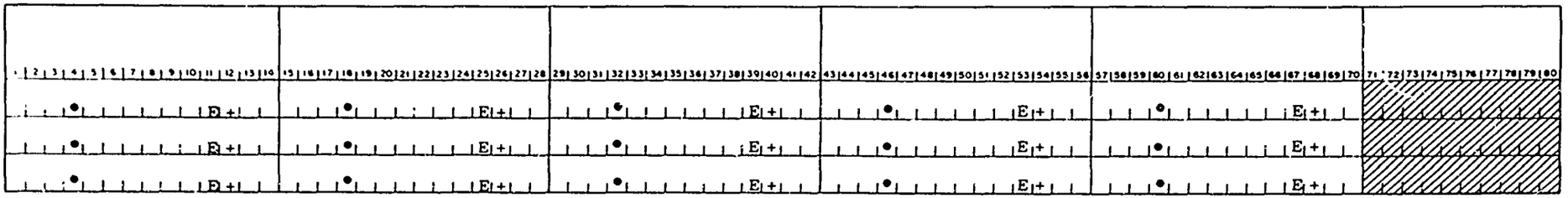
Energy Mesh (Omit if NEWL = 0. If NEWL > 0 enter NEWL + 1 energy bin boundaries in decreasing order)



GASP – (CONTINUED)

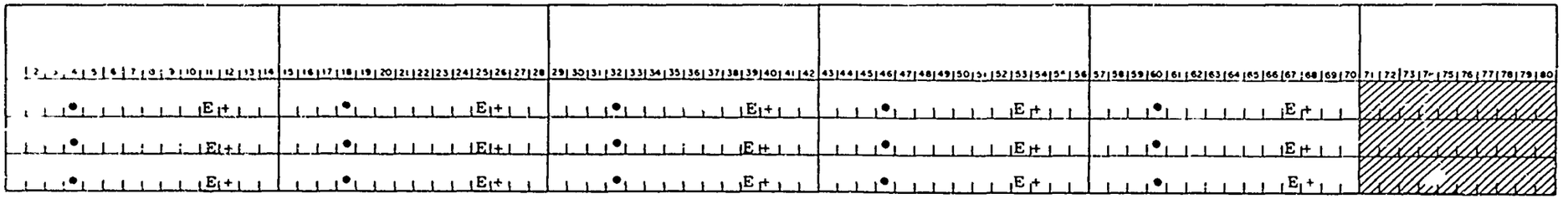
Page 4 of 4

Enter "NEW" Energy Weight Sets – Each set consisting of NEWL weights. Begin a new card for each set.



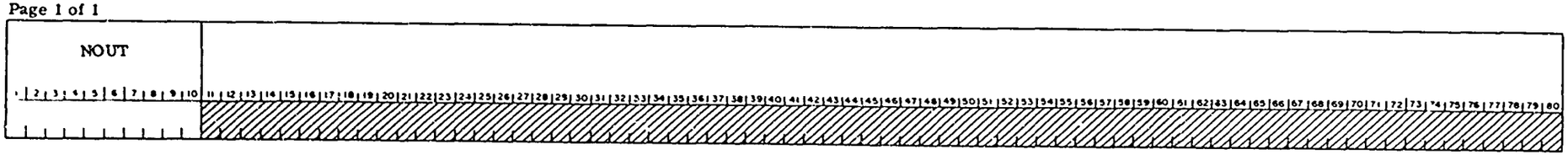
NOUT

Output Energy Bins. Enter NOUT + 1 energy bin boundaries in decreasing order.

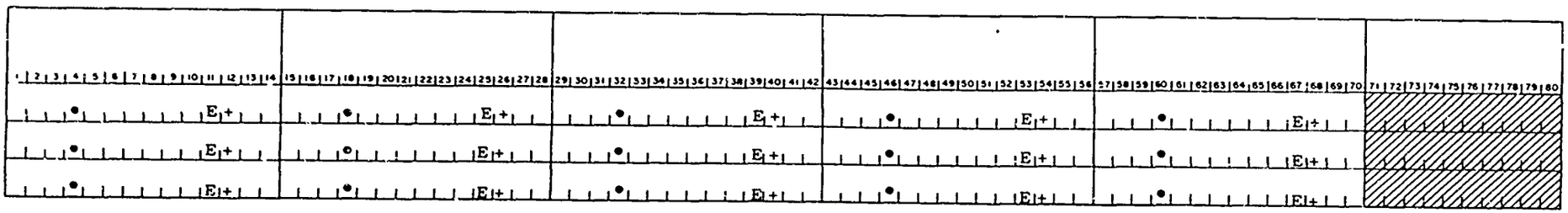


TRANSMIT INPUT

Number of Energy Bins for Edit of Source Tape Generated



Output Energy Bins. Enter NOUT + 1 energy bin boundaries, in decreasing order



14. REFERENCES

1. Eisenman, B. and Nakache, F. R. : UNC-SAM: A FORTRAN Monte Carlo System for the Evaluation of Neutron or Gamma-Ray Transport in Three-Dimensional Geometry, UNC-5093 (Aug. 31, 1964).
2. ADONIS – Advanced Shield Calculational Techniques, Vol. V, UNUCOR-635.
3. Kalos, M. H. and Goertzel, G.: Monte Carlo Methods in Transport Problems, "Progress in Nuclear Energy," pp. 315-369, Pergamon Press, New York, 1958.
4. Cranberg, L. et al.: Phys. Rev., 103:662 (1956).

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)* United Nuclear Corporation Research and Engineering Center Elmsford, New York 10523	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b GROUP

3 REPORT TITLE

UNC-SAM-2: A FORTRAN MONTE CARLO PROGRAM TREATING TIME-DEPENDENT NEUTRON AND PHOTON TRANSPORT THROUGH MATTER

4 DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Technical Report

5 AUTHOR(S) *(Last name, first name, initial)*

Troubetzkoy, Eugene, S.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1966	108	4

8a CONTRACT OR GRANT NO. DA-30-069-AMC-785(X) b. PROJECT NO. c. 2336 d.	9a. ORIGINATOR'S REPORT NUMBER(S) UNC-5157 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10 AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Army Ballistic Research Laboratories Aberdeen Proving Ground, Maryland

13. ABSTRACT

A description is given of a code treating time dependent neutron and photon transport through matter in complicated geometrical configurations.

DD FORM 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Transport of Neutron and Photon Monte Carlo Boltzmann Equation						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, *or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS. Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.